Chapter 7
System Assembly
(with Example Case 1)

Generally speaking, the construction of a virtual turbine in ADAMS/WT mimics the construction of a real turbine, in that the aggregate elements are first constructed separately, then placed in the correct positions and connected together. Many of these connections have been made more automatic in this version of WT. Finally, a few site- or turbine-specific "adjustments" are made before the machine is placed on-line.

To demonstrate the entire process, this section describes the construction, in ADAMS/WT, of an example 2-bladed horizontal-axis machine. The construction steps are listed below, then discussed in more detail in the following text. Following chapters will show how to enhance this machine with more flexible elements, etc., how to build a 3-bladed version and how to build a counter-clockwise, upwind machine.

7.1 Outline

NOTE: To work through this example, you should switch into the *nrel/examples/case_1* directory before starting ADAMS/View. Assuming you have set up the environment variables and *aview.pth* file correctly, you can then start View and load the ADAMS/WT overlay with the by reading in the command file *wt_main.cmd*. At this point you should be ready to begin *case_1*.

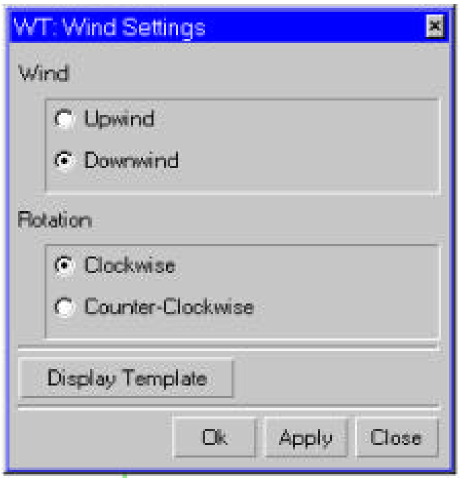
1. Set the direction of rotation and rotor orientation.
2. Create the tower aggregate element.
3. Create the nacelle aggregate element on top of tower.
4. Create the power train in the nacelle.
   a) Stator
   b) Low-speed Shaft
   c) Motor-Generator
1. Create the hub aggregate element.
2. Relocate the hub to the end of the low-speed shaft.
3. Create blade #1 and relocate to one end of hub.
4. Create blade #2 and relocate to other end of hub.
5. Add teeter stops and hub spring.
6. Add tip weights.
7. Add AeroDyn aerodynamics to each blade.
8. Add gravity.
9. Add desired output requests.
10. Create a user-executable version of Solver and do the analysis.
11. Look at the results.

NOTE: In order to avoid losing your work, we recommend that you save the ADAMS/View session to a binary file after each section in the example is completed. This can be done through the FILE SAVE menu, or from the command line by typing:

file binary write file=case_1 (or just *fi bi wr fi=case_1*)

7.2 General Set-Up

This first example uses the default set-up, that is downwind and clockwise. If you wish, you can manually select this from main WT menu using WIND/ROTATION SETUP. The internal ADAMS/View variable *dir_rot* will be set to the string "DC".



WT: Wind Settings

Wind

○ Upwind
● Downwind

Rotation

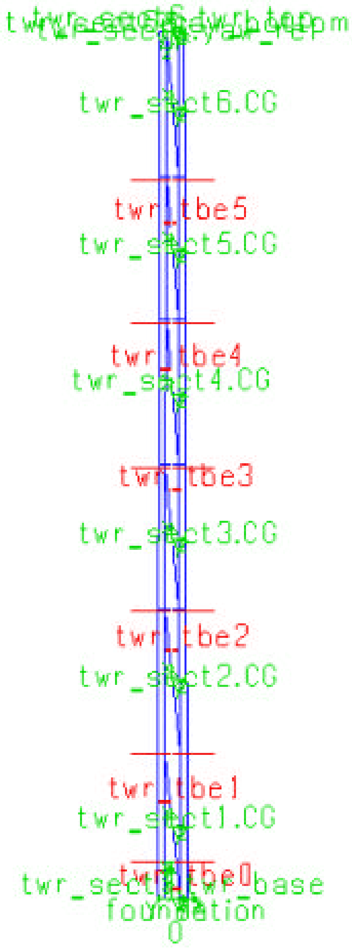
● Clockwise
○ Counter-Clockwise

7.3 Tower

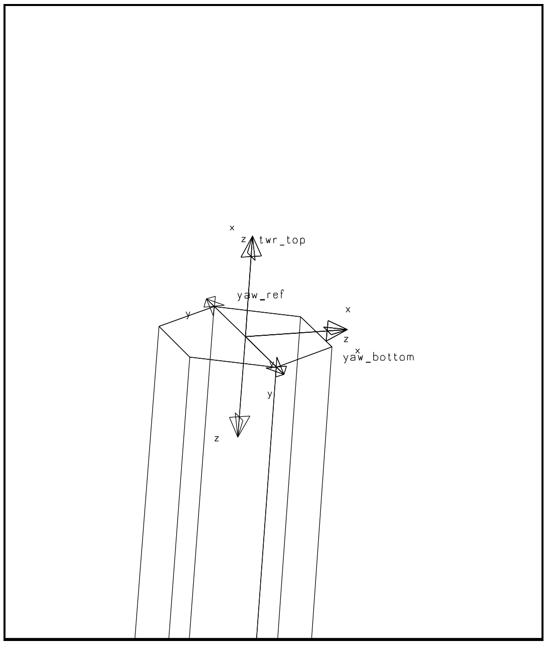
The input data for the case_1 tower are in *tower.dat* in the *examples/case_1* directory. Bring up the tower create panel and use the following values:

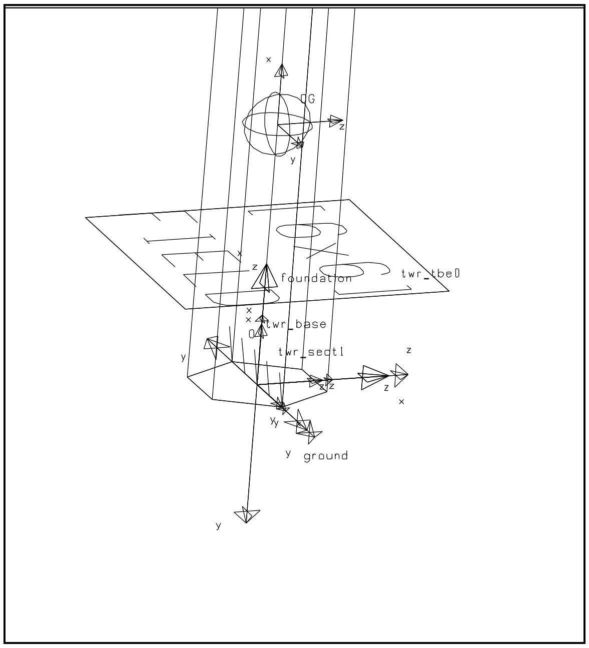
Number of Parts = 6
Tower Height = 26 m
Tower Properties File = tower.dat
Number of Sides = 6
Bottom Diameter = 1.0 m
Top Diameter = 0.8 m
Color = your choice

If everything is working properly, ADAMS/WT should display an information window which monitors the automatic tower construction. Depending on the speed of your platform, building the tower may take several minutes. When the macro terminates, the info window should disappear and you should see the tower.

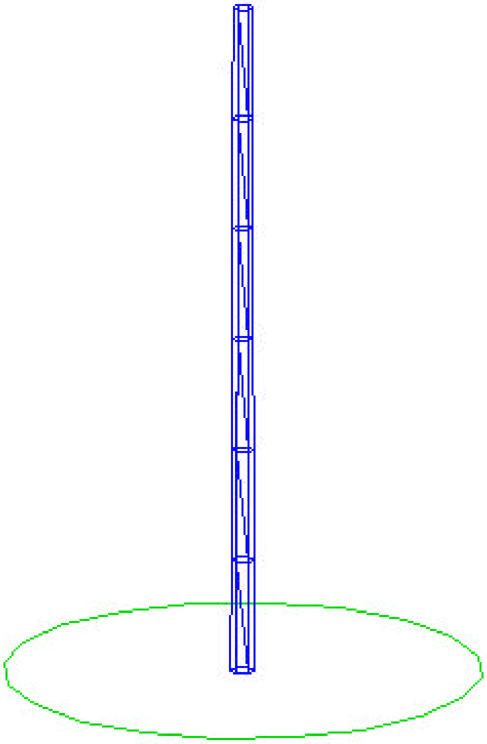


At the top of the tower is the *yaw_bottom* MARKER for later attachment to the nacelle. The bottom of the tower is a half-length tapered beam connected to the *foundation* MARKER on the ground. (These graphics are left over from version 1.5, but are the most clear I could get.)





If you desire, you can add some graphics to the *ground* PART to give some perspective during the subsequent modeling. Here you can see the effect of adding a 10-m radius circle graphic centered on the *O* MARKER:

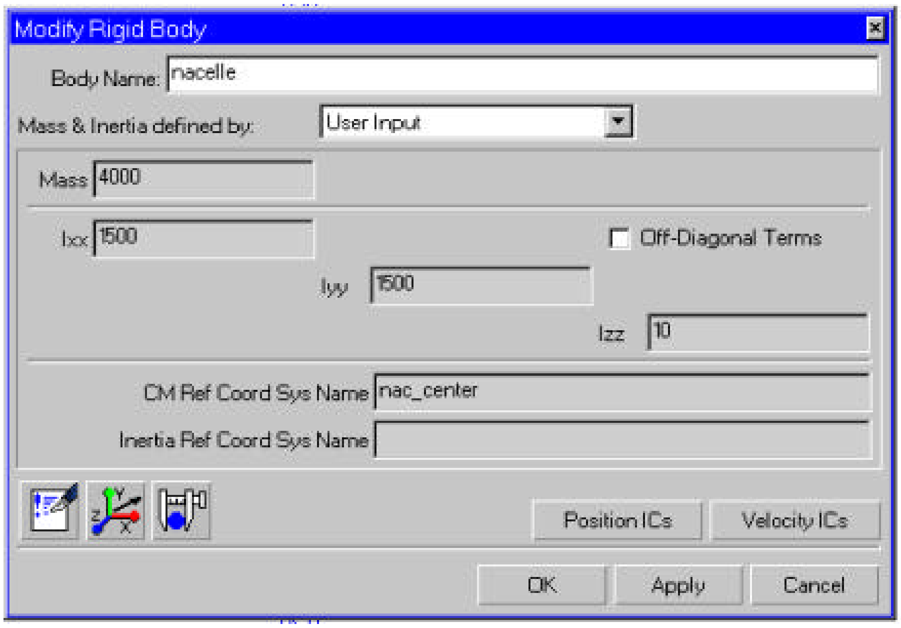


7.4 Nacelle

After completing the tower, we next move to the nacelle. For this example, you can use the following data in the NACELLE CREATE panel:

Yaw Type = free_yaw
Marker_on_Tower = yaw_bottom
Shaft_Height_Above_Bearing = 1.0 m
Yaw Stiffness = 0.26 N-m/deg
Yaw Damping = 2.6 N-m-sec/deg
Diameter_at_Bearing = 1 m
Upwind Length = 1 m
Upwind Diameter = 0.6 m
Downwind Length = 1.3 m
Downwind Diameter = 0.5 m

Then, bring up the NACELLE MODIFY panel to set the mass properties for the nacelle. When you hit the MASS PROPERTIES button, it will automatically bring up the standard View PART MODIFY panel for the *nacelle* PART. You should enter the values shown in this panel, leaving the other fields blank:



Modify Rigid Body

Body Name: nacelle

Mass & Inertia defined by: User Input

Mass 4000

Ixx 1500

☐ Off-Diagonal Terms

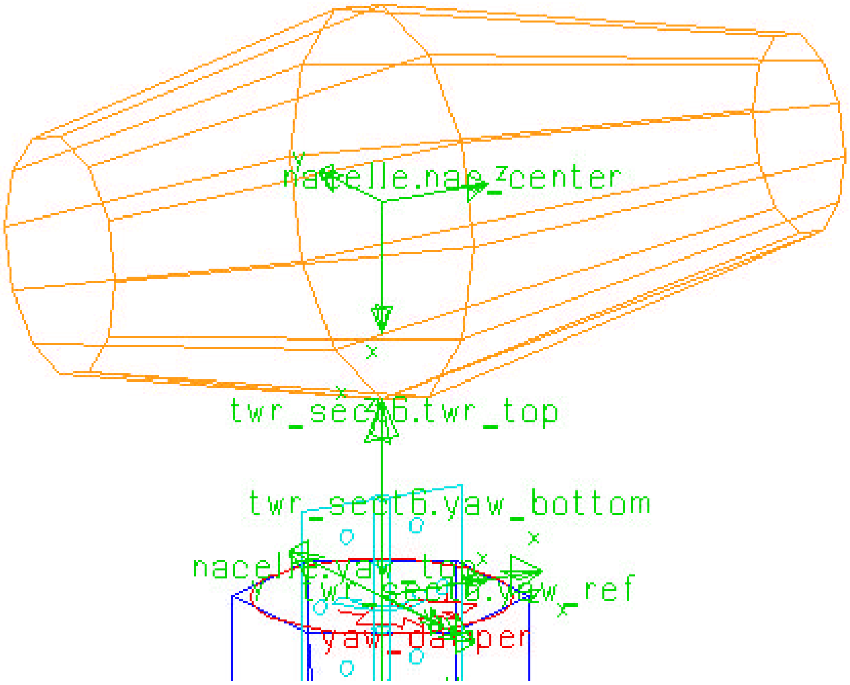
Iyy 1500

Izz 10

CM Ref Coord Sys Name nac_center

Inertia Ref Coord Sys Name

After completing the PART MODIFY panel (use OK), you should Close the NACELLE MODIFY panel without doing anything else. By picking on the nacelle and rotating a bit, you should see something like this:

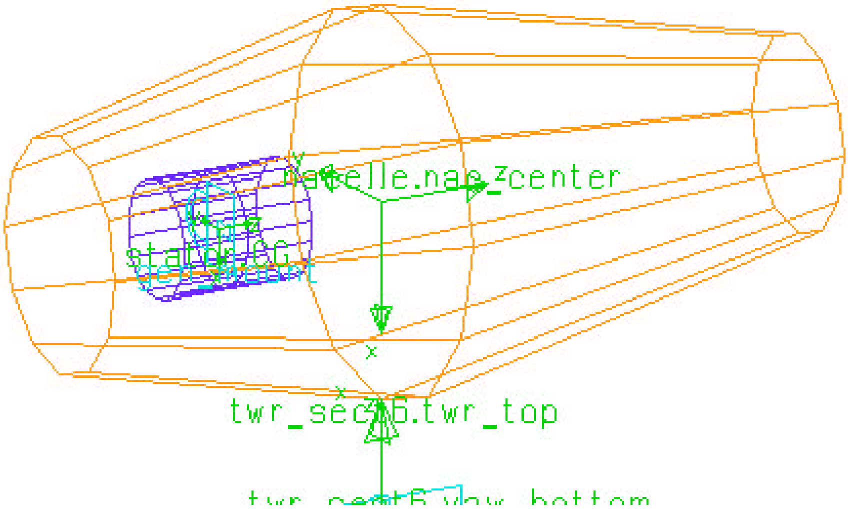


7.5 Power Train

Begin the power train by creating a generator body, called the *stator*. Remember that you can split the non-rotating inertia between the nacelle and the stator as you see fit. However, it is convenient to have a *stator* PART for connecting to one side of the motor-generator later on. Bring up the POWER_TRAIN GENERATOR_BODY panel and enter these values:

Location = 0,0,-0.5
Orientation = 0,0,0
Relative_to = nac_center
Mass = 200 kg
I_xx = I_yy = 20 kg-m^2
I_zz = 10 kg-m^2
Graphics Diameter = 0.3 m
Graphics Length = 0.4 m

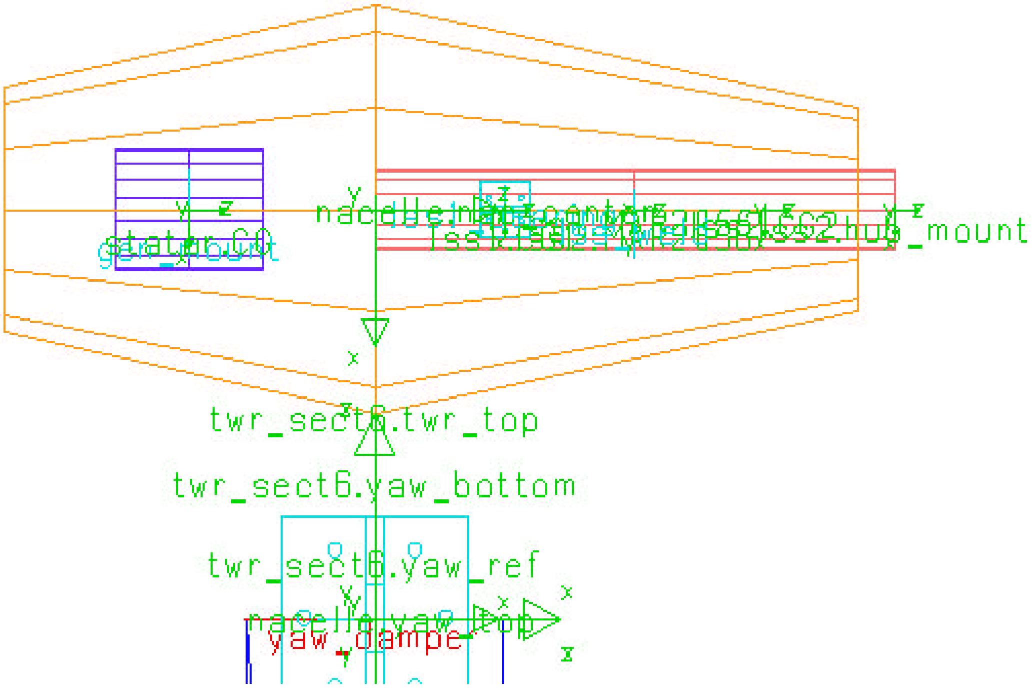
This should add the *stator* to the nacelle as shown here:



This example case does not include a high-speed shaft or gearbox. The low-speed shaft is connected directly to the motor-generator and the appropriately scaled inertia has been added to the two low-speed shaft parts. Also, this case uses a rigid shaft, created by using the torsion-only type of shaft and entering zero (0) values for the both shaft stiffness and shaft damping. Note that when you create a rigid shaft, WT will automatically “weld” the lss1 and lss2 shaft PARTs together with a FIXED-type JOINT, and will also remove the redundant *lss2_bearing* revolute JOINT. You may later need to relocate the *lss1_bearing* to a realistic position if you want to get more correct reaction loads on it.

Next, select the POWER_TRAIN LOW-SPEED_SHAFT TORSION_ONLY panel and enter the following values:

Location = 0,0,0.7 m
Orientation = 0,0,0
Relative_to = nac_center
Diameter = 0.2 m
Length = 1.4 m
Stiffness = 0
Damping = 0
Mass = 100 kg
Ixx (=Iyy) = 10 kg-m^2
Izz = 1 kg-m^2



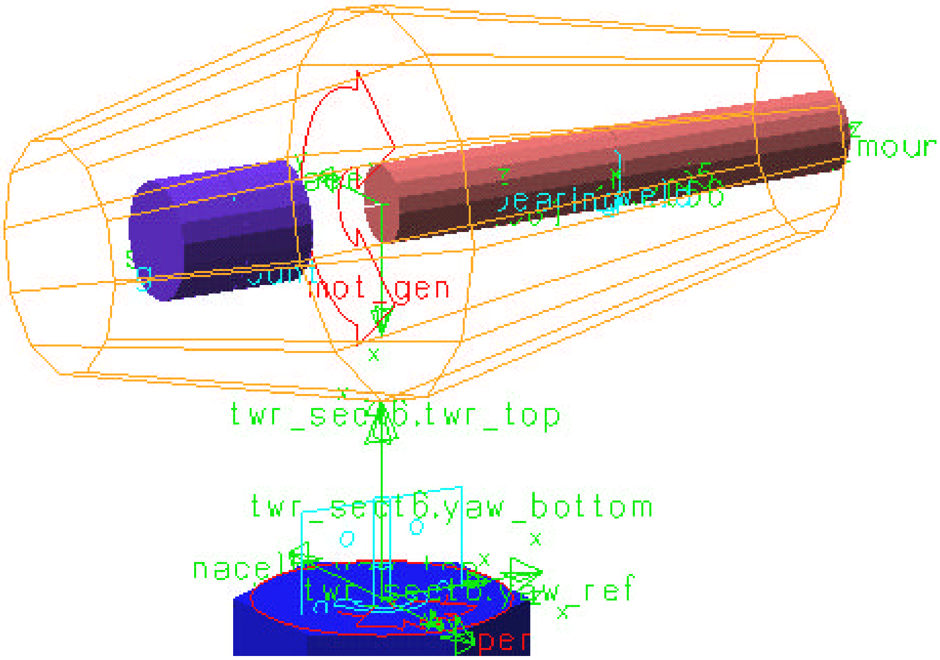
Since there is no high-speed shaft or gearing, you should now create the motor-generator. Here, the voltage is stepped up smoothly from zero to 240 volts over the first 200 msec of the simulation. This allows for a static solution, and for a more gentle startup. Bring up the POWER_TRAIN MOTOR-GENERATOR panel and enter these values.

Line Voltage = STEP(TIME,0,0,0.2,240)
Desired Speed (rpm) = 60
I Marker = .hawt.lss1.CG1
J Marker = .hawt.stator.CG

The torque-voltage-speed relation in the TORQUE_FUNCTION entry is described for this case using Thevenin’s equation, which is automatically generated by filling in these coefficients and hitting the Generate Torque Function button:

A_0 = 0.0128067
C_0 = 0.000157
C_1 = 0.00106
C_2 = 0.02428

Executing the motor-generator panel will create the *mot_gen* rotational SFORCE element which acts like the real motor-generator. The completed power train look like:



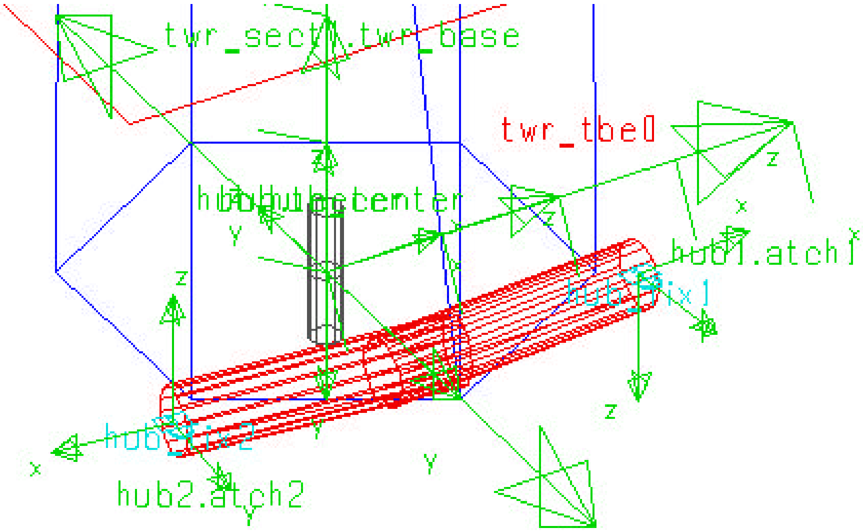
7.6 Hub

ADAMS/WT has four options for rotor hubs, 2-bladed teetering and 3-, 4- and 5-bladed rigid. For any of the hubs, you can later add flexibility between the blade attachment and the hub itself using the HUB MODIFY panels. For the case_1 example, we will create a two-bladed, teetering hub using the ROTOR_HUB CREATE 2-BLADED_TEETER panel. When you select this option, WT will display a template which makes it easier to visualize exactly to what the various parameters refer.

You should create a 2-bladed teetering hub using the following parameters:

Precone = 5 deg
Axial_Offset = -0.3 m (underslung)
Delta_3 = 0 deg
Root_Cutout = 0.5 m

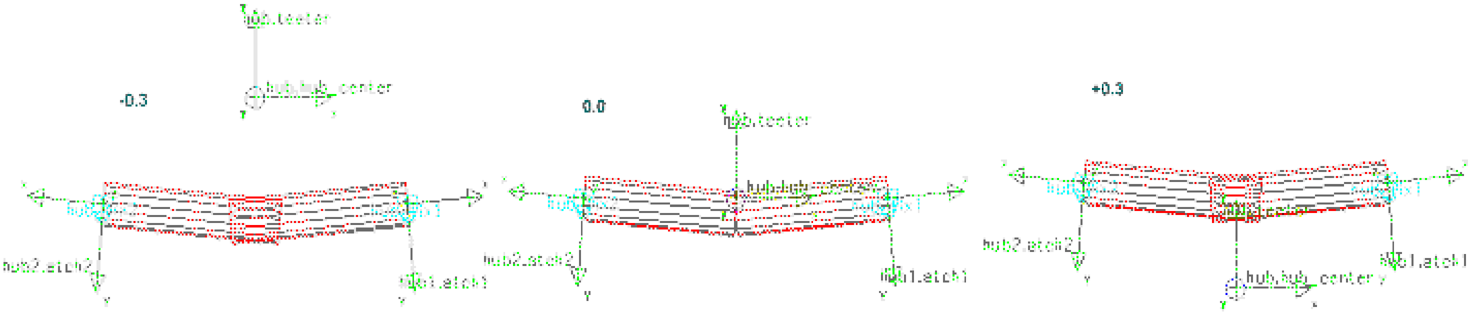
This will build the *hub* PART geometry, but leave it at the global origin (base of the tower).



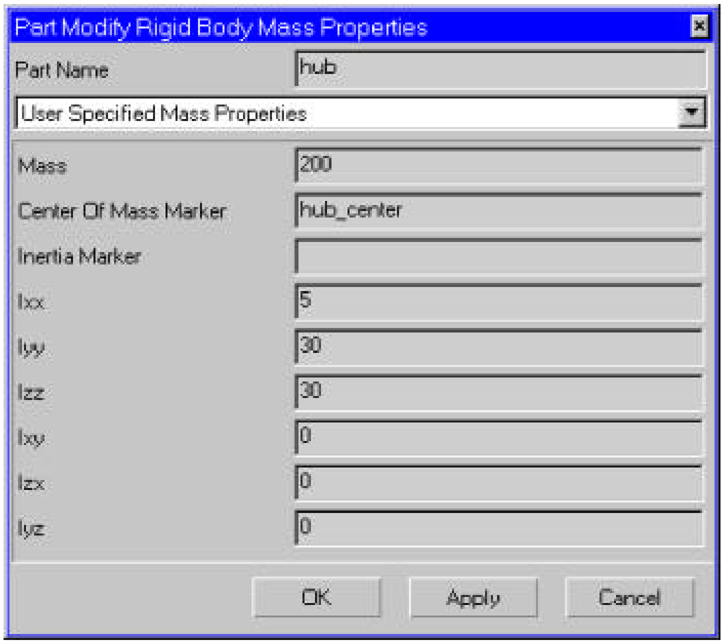
It might be helpful here to examine more closely exactly what happens when you change the axial_offset parameter and how it is related to the precone and root_cutout parameters. The following figure shows the same hub you just created, but using -0.3, 0.0 or +0.3 as the axial_offset. In each window, the teetering hinge line is centered in the frame. The axial_offset is shown as the vertical distance between the hinge line (*teeter*) and the blade attachment points (*atch#*).

Note that the actual location of the bottom of the rotor cone, that is, the imaginary intersection of the two blade axes, is <u>not</u> at axial_offset below the hinge line, unless the precone is 0. This is clearly shown in the center frame, where the axial_offset is 0 and precone is 5°, and the intersection is still below the hinge. In fact, the distance from the hinge point to the “bottom” is:

$$axial\_offset - root\_cutout*\cos(precone)$$



After creating the *hub* PART, you need to bring up the ROTOR_HUB MODIFY 2-BLADED_TEETER panel, hit the MASS_PROPERTIES button and enter these data:



Part Name: hub

User Specified Mass Properties

Mass: 200

Center Of Mass Marker: hub_center

Inertia Marker:

Ixx: 5

Iyy: 30

Izz: 30

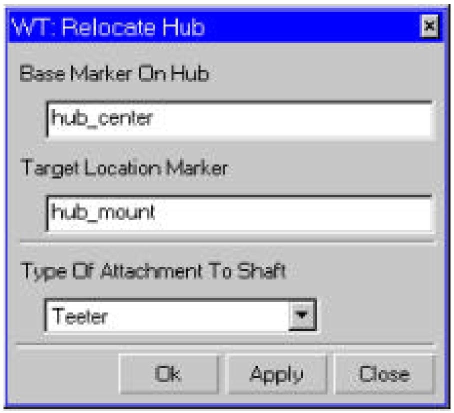
Ixy: 0

Izx: 0

Iyz: 0

After executing this panel, you will be returned to the ROTOR_HUB MODIFY 2-BLADED_TEETER panel. You should Close out of the hub modify panel without making any other changes.

Next, you need to relocate the *hub* to the end of the low-speed shaft, using the ROTOR_HUB RELOCATE panel with the entries shown:



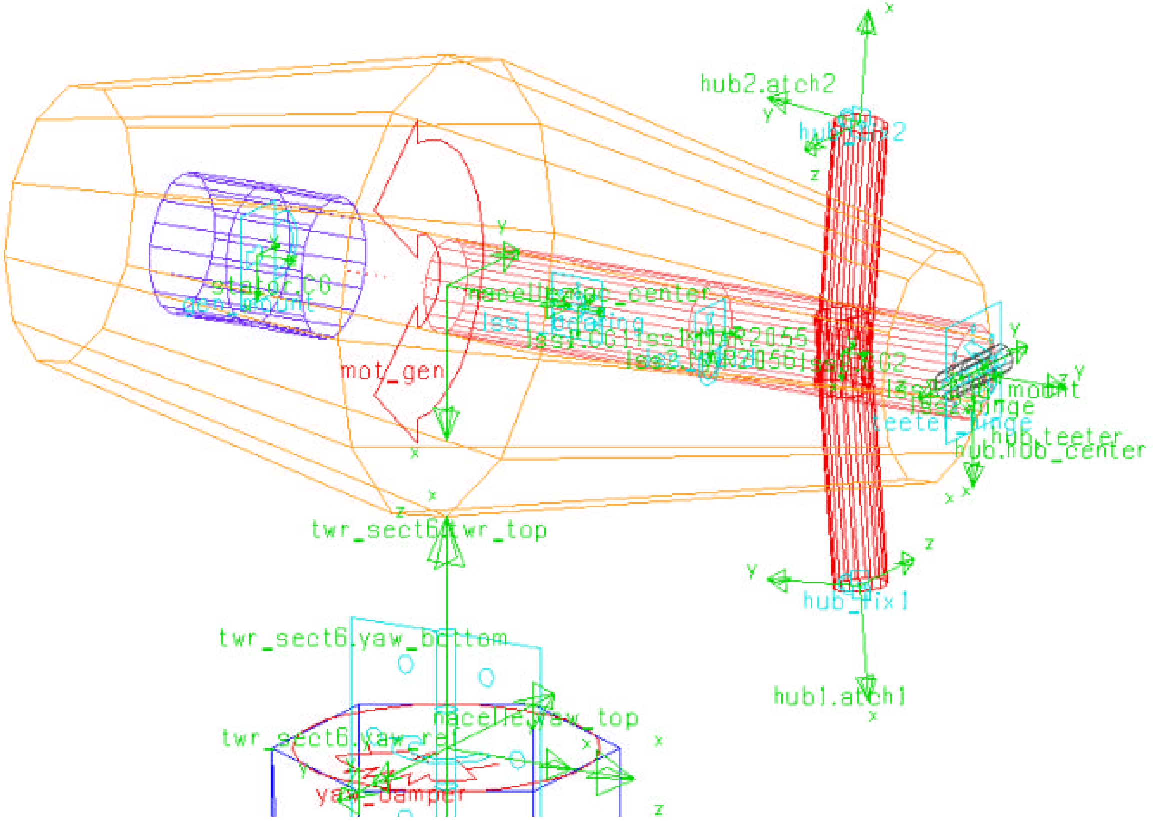
Base Marker On Hub: hub_center

Target Location Marker: hub_mount

Type Of Attachment To Shaft: Teeter

This will both move the *hub* to its correct position and create the *teeter_hinge* revolute JOINT between it and the *lss2* PART at the correct position and orientation. Note that using these default MARKERs for the relocation means that the teetering hinge will be located exactly at the very end of the low-speed shaft. To change this in another model, you should move the *hub_mount* before you relocate the *hub*.

Note that you can change the size of the *teeter_hinge* icon using the CONSTRAINT ATTRIBUTES panel (starting from the Command Navigator). In this case, a size of about 0.25 meters seems to look pretty good.

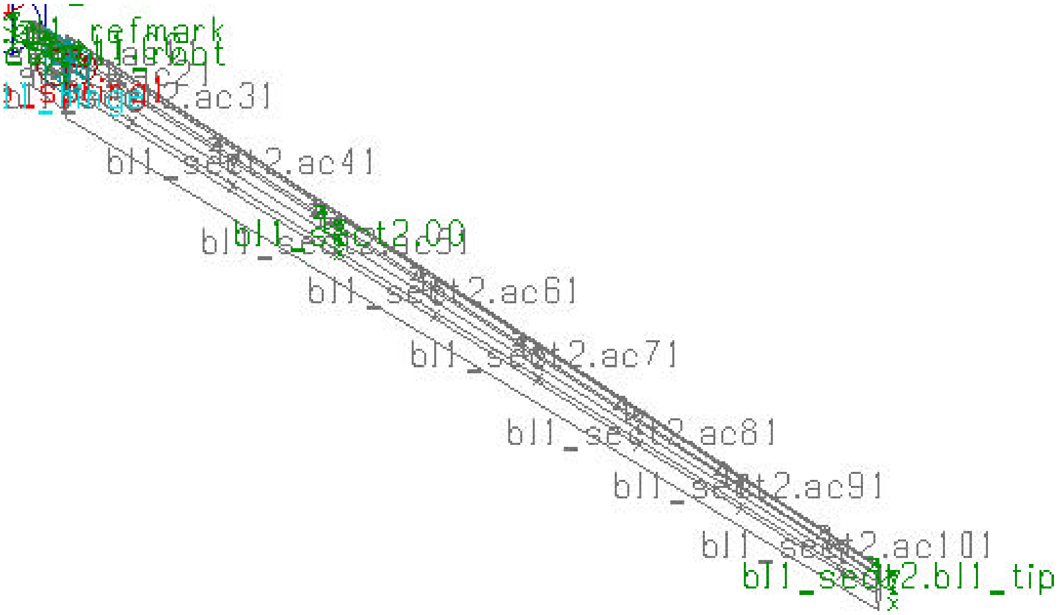


7.7 Blade #1

This example uses a rigid/hinged blade. The blade data for case 1 is found in the *blade.dat* file in the *examples/case_1* directory. Bring up the rigid blade creation panel, ROTOR_BLADE CREATE RIGID_BLADE STRUCTURAL, and use the following values:

Blade Number = 1
Aero Sections Inboard = 2
Aero Sections Outboard = 8
Property File = blade.dat
Tip Radius = 13.0 m
Hinge Radius = 1.0 m
Root Radius = 0.5 m
Hinge Stiffness = 100 N-m/deg (dummy value)
Hinge Damping = 10 N-m-sec/deg (dummy value)
Fixed # of Aero_Points per Section = 1

If everything is working, ADAMS/WT will run the auxiliary program *wtblade*.exe and should display an information window which monitors the blade construction. This could take a couple of minutes. The blade, like the hub, is originally constructed on the ground. Note that the aerodynamic center locations are already in place (the *ac##* MARKERs) and the tip is also marked for later attachment of the tip weight.



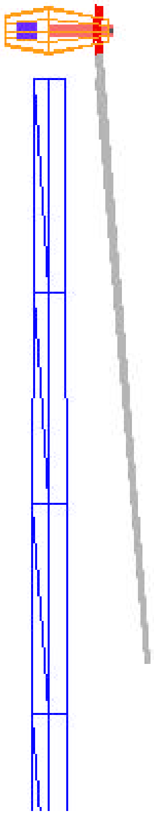
After this, you of course need to relocate the blade to the correct attachment point on the hub. Using the ROTOR_BLADE RELOCATE panel with the following values will both move the blade to the *atch1* MARKER on the hub and connect it to the hub with a FIXED-type JOINT called *bl1_rootweld*.

Blade Number = 1
Base Marker on Blade = bl1_root
Target Location Marker = atch1
Pitch Angle = -5.68 deg

Note that this value for pitch angle, -5.68°, should give very similar response to the -10.8° values used in example cases 2 and 3 with flexible blades. Checking the *blade.dat* file, you can see that the aerodynamic pitch at the blade root is approximately -5.12°. Referring to the description of blade relocation in Section 5.6.4, we find that the expression for pitch setting for a rigid blade is:

rigid_pitch_setting = *measured_pitch_angle* - *root_twist_angle*

which in this case gives (-10.8 - -5.12) = -5.68. Note also that, by convention, the #1 blade begins in the zero azimuth position, which is pointing straight down. The following graphic has the ADAMS/View icons turned off for clarity.



7.8 Blade #2

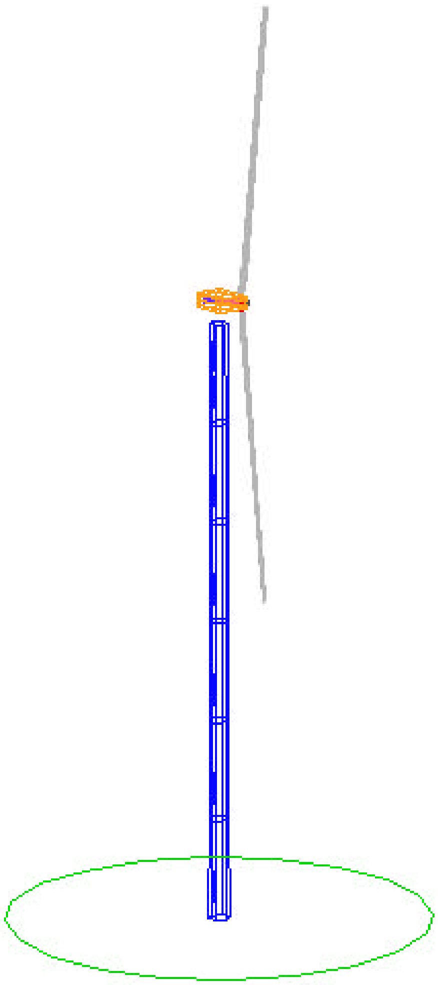
The construction of blade number 2 is identical in every way (except numbering) to blade number 1. It uses the same blade properties file and the same number of aerodynamic sections and markers as blade #1. Note that the ROTOR_BLADE CREATE RIGID_BLADE STRUCTURAL panel “remembers” the data from the first blade, so you need only input the number 2 in the appropriate field.

Blade Number = 2

After creating the #2 blade, you should again use the ROTOR_BLADE RELOCATE panel, but with the following values:

Blade Number = 2
Base Marker on Blade = bl2_root
Target Location Marker = atch2
Pitch Angle = -5.68 deg

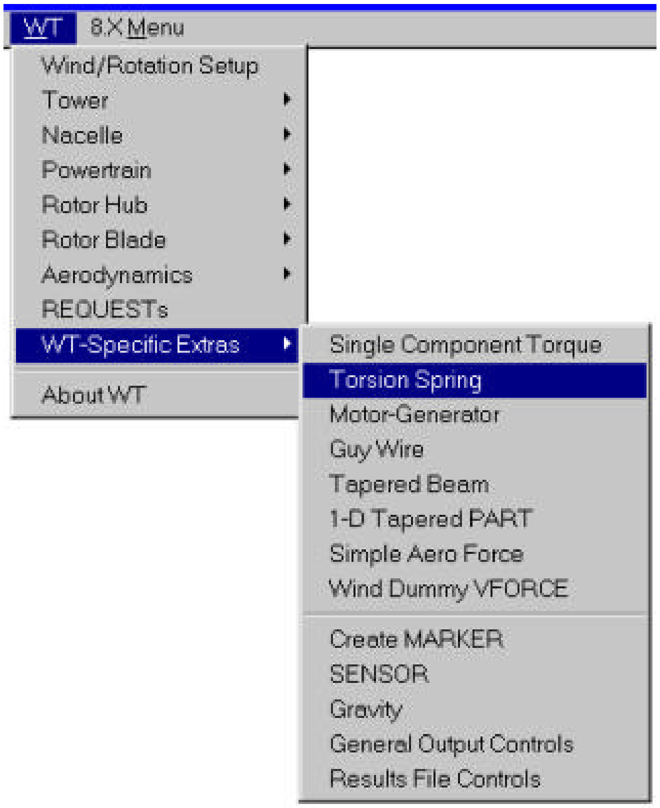
After the second blade is attached to the hub, it should appear as follows:



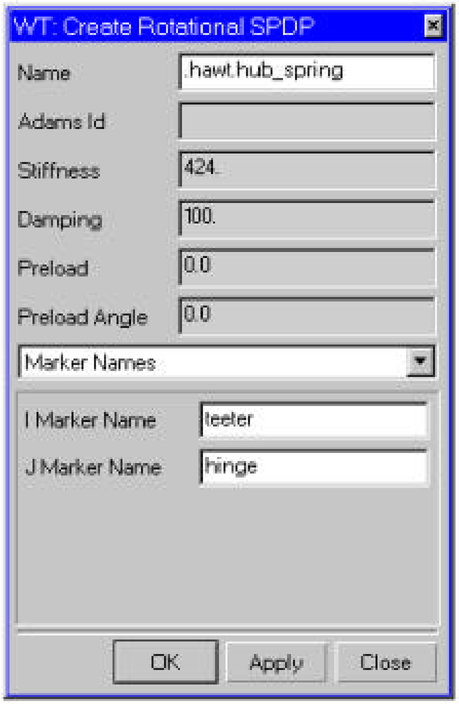
7.9 Teetering Spring and Stops

The example rotor has a centering spring in the hub, in addition to the teeter stops. Both of these are modeled in ADAMS as force elements attached to the same two markers which define the teetering pin joint, *teeter_hinge*. Those are automatically created by ADAMS/WT when the hub is relocated to the end of the shaft and consist of a MARKER named *hinge* on the *lss2* PART and the *teeter* MARKER on the *hub* PART.

For the centering spring, you should create a SPRINGDAMPER named *hub_spring* using the WT-Specific Extras / Torsion Spring panel from the WT main menu structure.



This brings up the SPDP panel, which you fill in as shown:



WT: Create Rotational SPDP

Name: .hawt.hub_spring

Adams Id:

Stiffness: 424.

Damping: 100.

Preload: 0.0

Preload Angle: 0.0

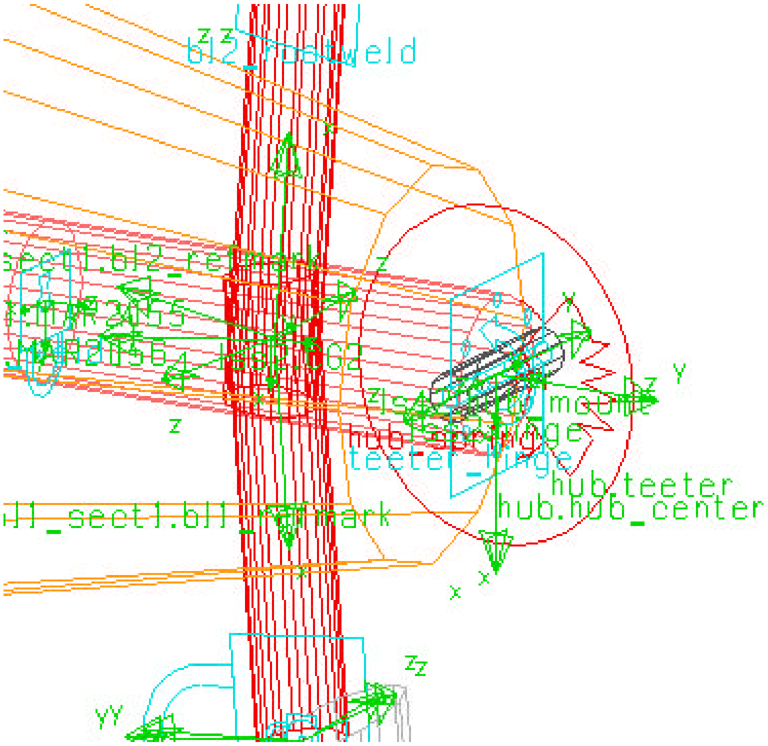
Marker Names

I Marker Name: teeter

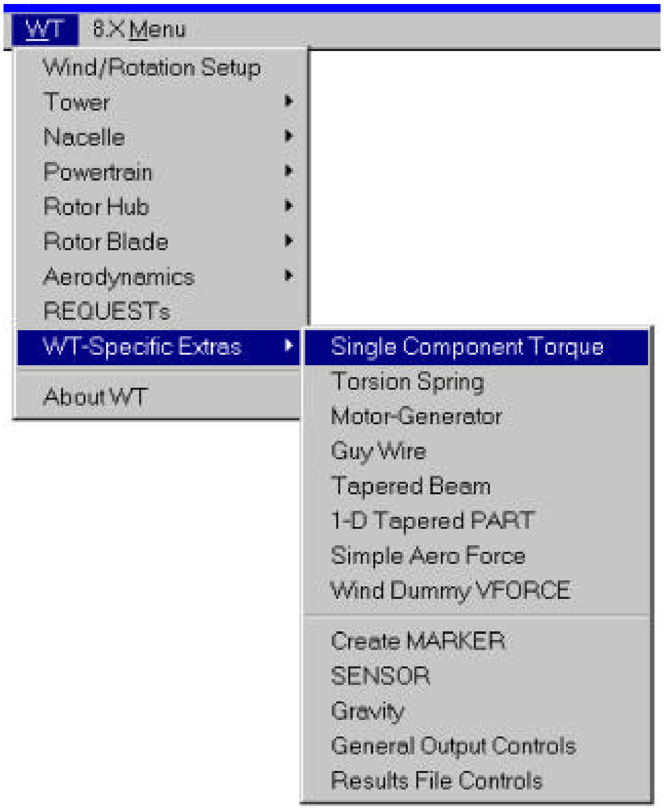
J Marker Name: hinge

Note that even though this input panel uses stiffness values per degree and damping values per degree per second, when the actual ADAMS dataset file (*.adm*) is written, these values will be automatically converted to per radian. The values will remain in per degree in the A/View command (*.cmd*) file. All calculations internal to ADAMS/Solver are done in radians.

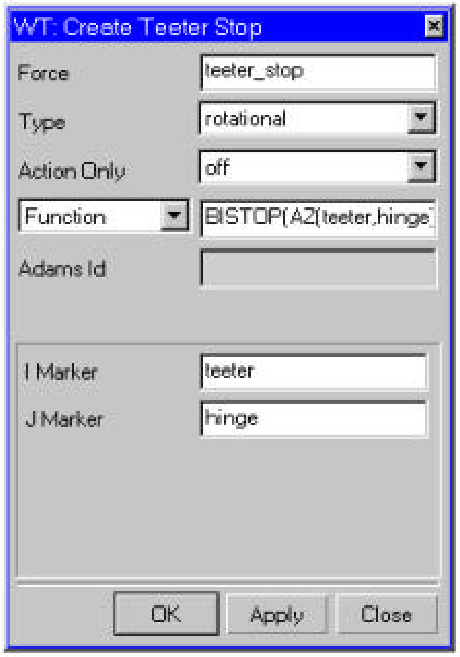
After scaling down the icons for convenience, the hub center should look like:



It turns out to be very simple to model the teetering stops in ADAMS with a single rotational SFORCE element containing the BISTOP function, which is designed to model two-ended slots. You should create this SFORCE named *teeter_stop* on the same two MARKERs as the *hub_spring* and *teeter_hinge*. This is done using the WT-Specific Extras / Single Component Torque panel from the WT main menu structure.



This brings up the Teeter Stop panel which you should fill-in as shown:



WT: Create Teeter Stop

Force: teeter_stop

Type: rotational

Action Only: off

Function: BISTOP(AZ(teeter,hinge

Adams Id:

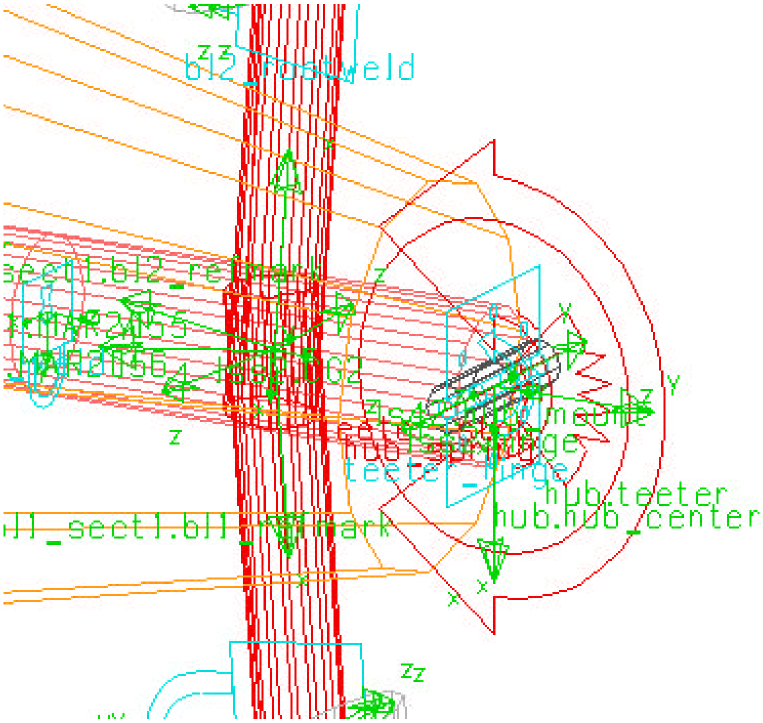
I Marker: teeter

J Marker: hinge

The exact text for the teetering stop function expression is given here. You can easily enter this in the panel by right-clicking in the function field to bring up the Function Builder tool and then entering it there.

```
BISTOP(AZ(teeter,hinge),
       WZ(teeter,hinge,hinge),
       -.044,.044,683000,1,8430,.001)
```

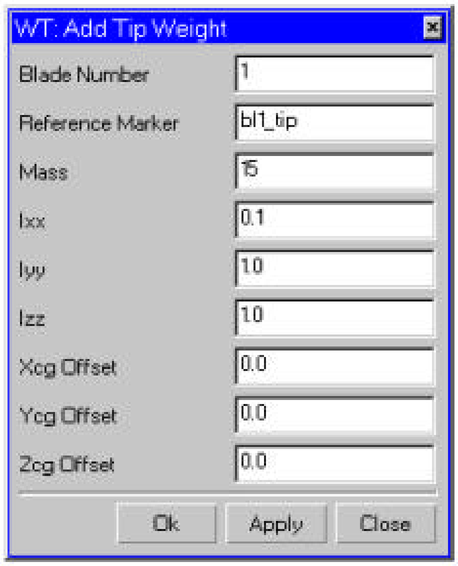
Please break the expression onto three lines as shown. If you put it all on one line, it will be too long for View to handle internally after the MARKER names are fully expanded. Note also, that in this case, because you are writing an ADAMS/Solver function expression that will be transferred directly into the dataset, the stiffness and damping values are given per radian. You may again need to scale the force icon for aesthetics to get something that looks like:



7.10 Tip Weights

This example case uses tip weights on both blades. Tip weights are often used to give the inertial effects of an undeployed tip brake mechanism. This version of ADAMS/WT does not include deployable tip brakes as an automatically-generated aggregate element. You could, of course, add such a mechanism to each blade manually.

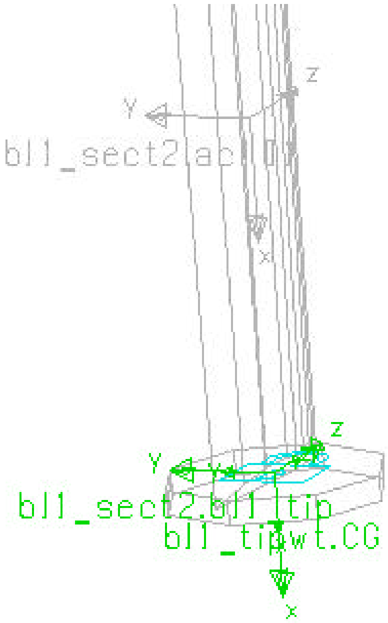
To add a tip weight to blade #1, bring up the ROTOR_BLADE ADD_TIP_WEIGHT panel and use the parameters shown:



WT: Add Tip Weight

Blade Number	1
Reference Marker	bl1_tip
Mass	15
Ixx	0.1
Iyy	1.0
Izz	1.0
Xcg Offset	0.0
Ycg Offset	0.0
Zcg Offset	0.0

This should change the blade tip to look like this:



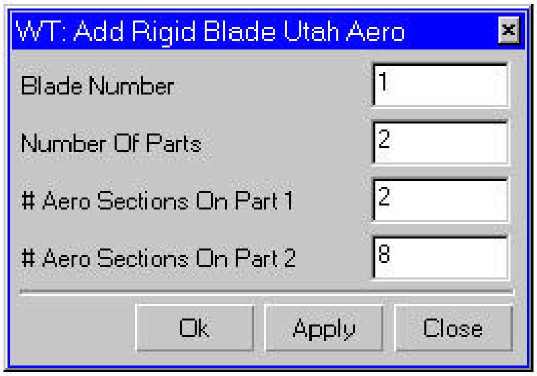
After applying the panel for blade #1, you should change the blade number and reference marker for blade #2 and apply again. WT should retain the values in the other fields.

Blade_number = 2
Ref_marker = bl2_tip

7.11 Aerodynamics

It is surprisingly simple to add aerodynamics to the model, due mainly to the large amount of up-front work done by Craig Hansen's group at the University of Utah, and the automation provided by ADAMS/WT. The AeroDyn aerodynamics subroutines are described in more detail in Appendix H. **This version of WT is designed to work with version 11.X of AeroDyn and will not work with earlier versions.**

To add the GFORCE elements which apply the aerodynamic forces computed in the AeroDyn routines to blade #1, you should bring up the AERODYNAMICS AERODYN_AERO RIGID_BLADE panel and enter the data shown:



WT: Add Rigid Blade Utah Aero

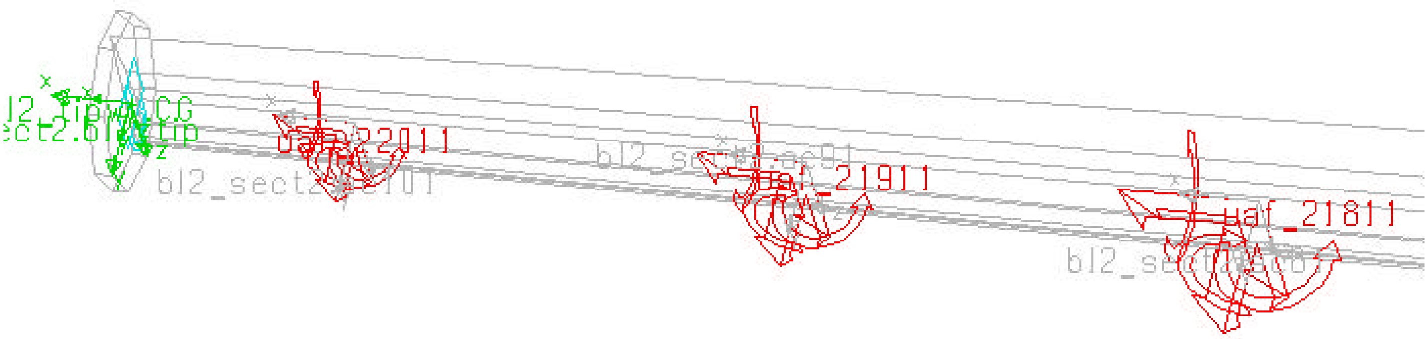
Blade Number: 1

Number Of Parts: 2

Aero Sections On Part 1: 2

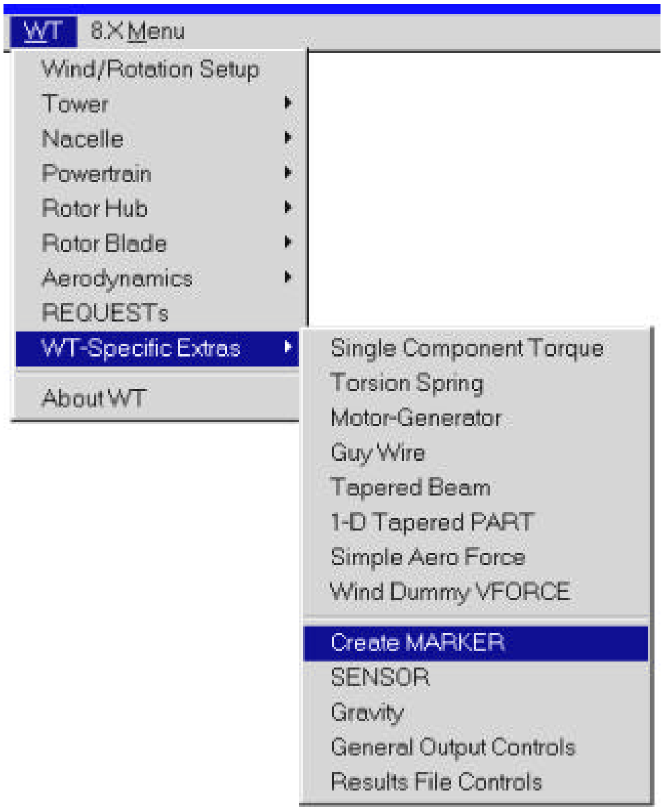
Aero Sections On Part 2: 8

Apply this panel for blade #1 and then change only the Blade_Number field (to 2) and apply the panel again for blade #2. Each time you apply the panel, A/View will spend some time doing computations and, if you have the standard command window open, you should see a series of messages flash past in the dialog window, like "The floating marker FMA119110 has been created on the part .hawt.ground." After the aerodynamic GFORCE elements are added, it will be nearly impossible to make out anything on the whole model when the View icons are turned on. By itself, a section of blade #2 would look like:

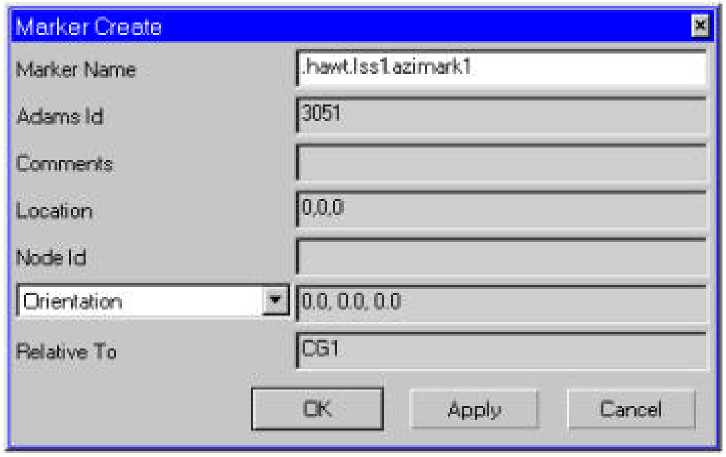


Before you can use the aerodynamics, you **must** add two specific MARKERs to the one of the low-speed shaft PARTs which AeroDyn will use to identify the blade azimuthal position. These MARKERs must have the ADAMS/Solver identifiers of 305#, where # is the blade number. Also they must be aligned such that their z-axes are along the shaft axis of rotation and their x-axes point radially outward in the plane formed by the shaft and matching blade axes.

To do the first marker, you can go through the A/View command navigator menus to open the MARKER CREATE panel, or just use the WT-Specific Elements menu.



You should fill out the panel as shown here:

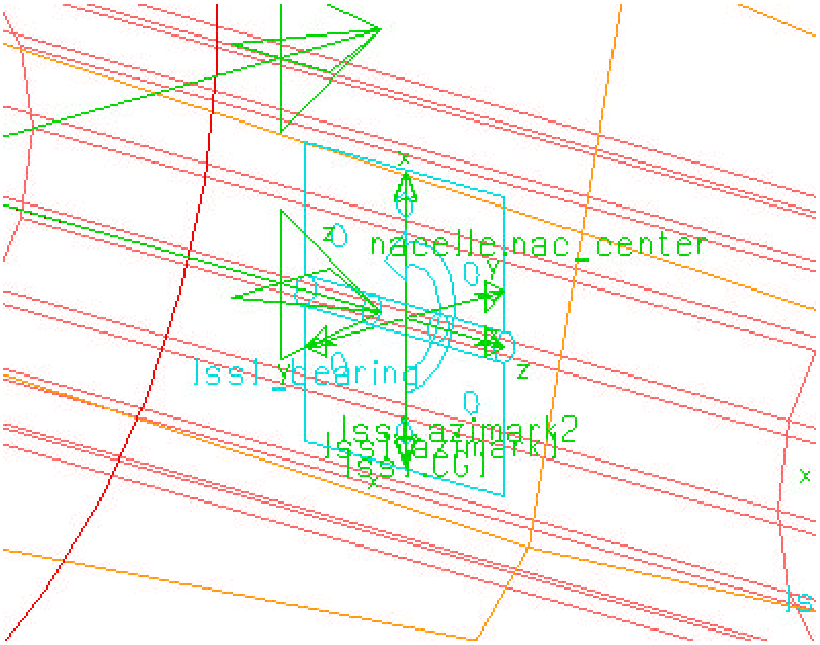

Marker Create

Marker Name	.hawt.lss1.azimark1
Adams Id	3051
Comments	
Location	0,0,0
Node Id	
Orientation	0.0, 0.0, 0.0
Relative To	CG1

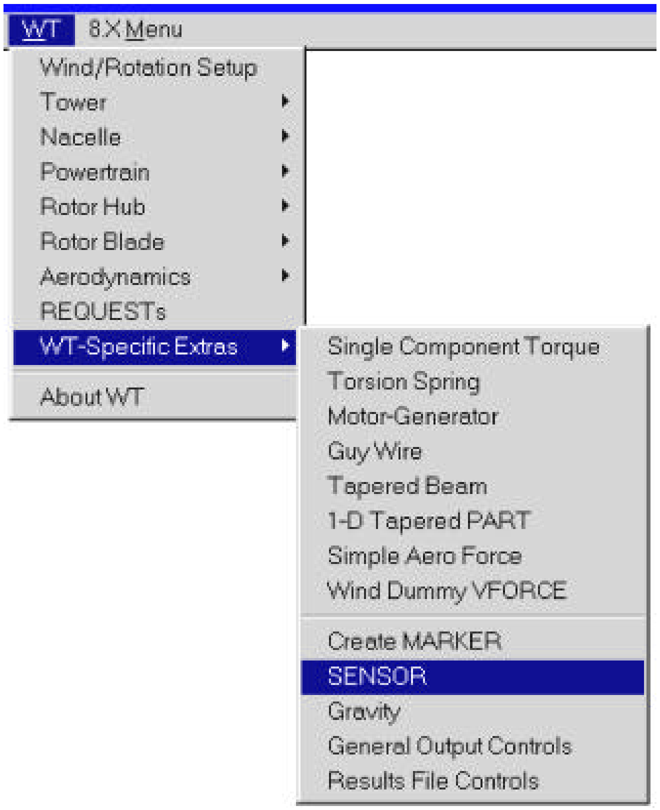
Apply the panel then re-do it with this information for the other blade:

marker_name = .hawt.lss1.azimark2
adams_id = 3052
location = 0,0,0
orientation = 0,0,180
relative_to = CG1

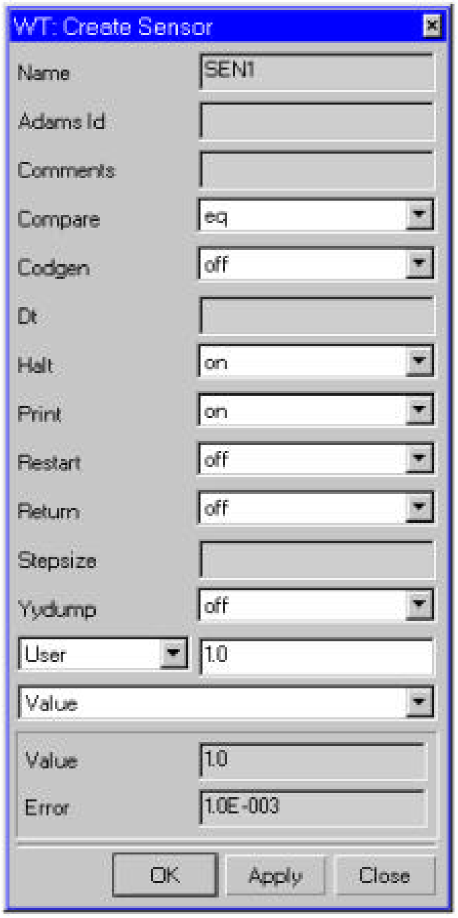
After creating both of these MARKERs, you can zoom tightly in on the *lss1_bearing* area and you should see there:



You must also create a dummy ADAMS SENSOR element to be used by the AeroDyn routines to monitor the progress of the simulation and advance the aerodynamic algorithm in step with it. This is also done through the WT-Specific Elements menu:



This should bring up a SENSOR creation panel with all the correct values filled in for you. You can check it against this one.



WT: Create Sensor

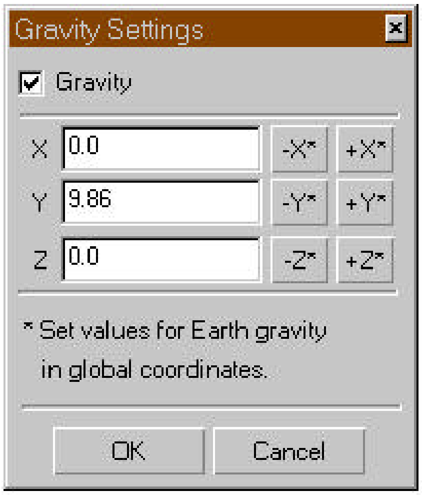
Name	SEN1
Adams Id	
Comments	
Compare	eq
Codgen	off
Dt	
Halt	on
Print	on
Restart	off
Return	off
Stepsize	
Yydump	off
User	1.0
Value	
Value	1.0
Error	1.0E-003

You also need appropriate input files for AeroDyn. These are the *yawdyn.ipt* and *airfoil.dat* files which can be found in the *examples/case_1* directory. You might note that the *yawdyn.ipt* file for case_1 is slightly different than the ones used for cases 2, 3 & 4. This is because that file includes data on the chord and spanwise length of each aerodynamic segment in the blade. These are different for the rigid blades and the flexible blades, which do not have the exact same radial segment spacing.

Finally, before you can actually use the rotor you have created here, you must create a user-executable version of ADAMS/Solver which includes the AeroDyn subroutines. This will allow you to run the model with these aerodynamics. How to do this and then how to run the model is covered in section 7.13 - Doing the Analysis.

7.12 Gravity

Gravity should be added to the model using the normal ADAMS/View menus (Settings / Gravity) or from the WT-Specific Extras menu, which also has a gravity choice. Complete the panel as shown:


Gravity Settings

☑ Gravity

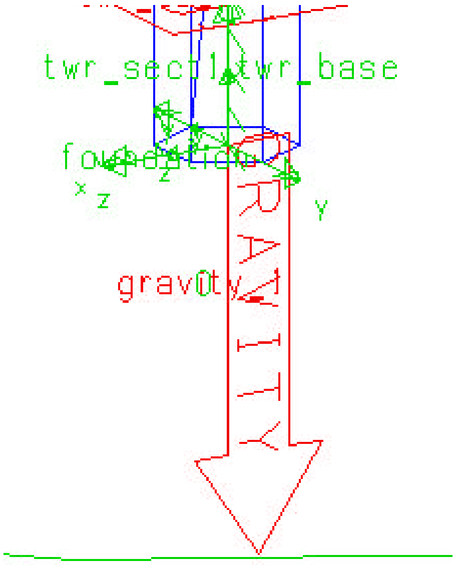
X 0.0

Y 9.86

Z 0.0

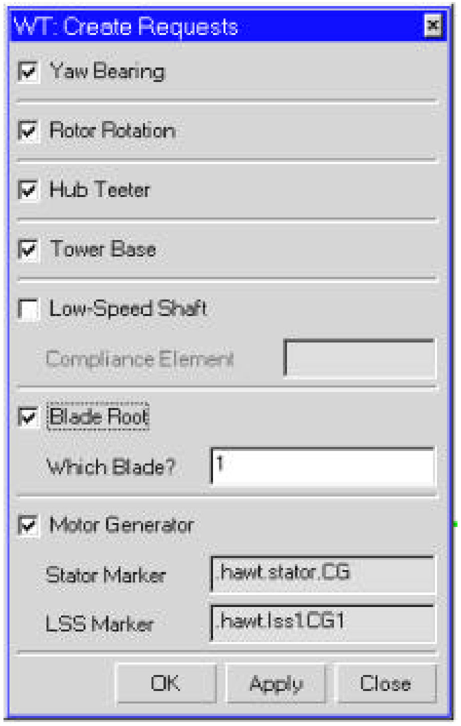
* Set values for Earth gravity in global coordinates.

The gravity force should appear as a small arrow at the base of the tower, pointing straight down (along positive global y axis).



7.13 Output Requests

Basic requests for tabular output are now automated in ADAMS/WT version 2.0. To turn on these outputs, which will show up in the *.req* file, go back to the main WT menu and select REQUESTS to see the various different kinds of output you can solicit from ADAMS, most of which only need a confirmation to be included.



☑ Yaw Bearing

☑ Rotor Rotation

☑ Hub Teeter

☑ Tower Base

☐ Low-Speed Shaft

Compliance Element

☑ Blade Root

Which Blade? 1

☑ Motor Generator

Stator Marker .hawt.stator.CG

LSS Marker .hawt.lss1.CG1

Note that for the MOTOR-GENERATOR request, you must define the two MARKERs between which the *mot_gen* force acts.

M-G attachment marker on stator = .hawt.stator.CG
M-G attachment marker on LSS = .hawt.lss1.CG1

For the BLADE_ROOT request, you must specify which blade you want to monitor.

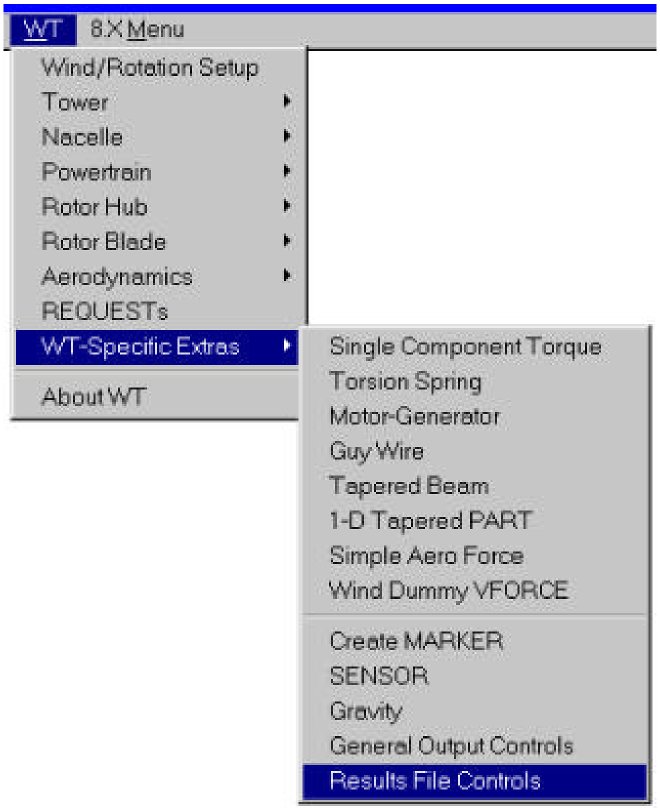
Which blade? = 1

The LOW-SPEED-SHAFT request is only used for flexible (either torsion-only or fully-flexible) and cannot be used for this example.

You may, of course, add additional REQUESTs to the model using any and all of the methods allowed by ADAMS, i.e. standard type requests, functionally defined requests or REQSUB user-subroutine-generated requests. AeroDyn comes with an example REQSUB user-subroutine which is described in detail in the AeroDyn appendix.

As in previous versions of ADAMS/WT, to save both space and run time, we recommend that you turn off the RESULTS output and create specific REQUESTs for any data you need. This minimizes the absolute amount of output and speeds the runs significantly. If you do need to use the results (.res) file, writing it UNFORMATTED will be both faster and produce much smaller files than the default FORMATTED output, but files will not be portable across platforms.

The Results control panel can be accessed from the WT-Specific Extras menu.

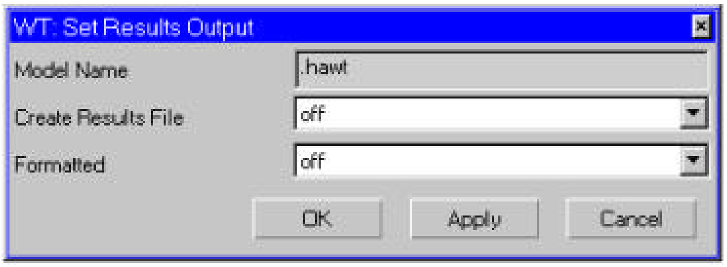


To turn off the results completely, set:

create_results_file = off
formatted = off

To keep the results file, but use the faster unformatted output, on the same panel you should set :

create_results_file = on
formatted = off


WT: Set Results Output

Model Name: .hawt
Create Results File: off
Formatted: off

7.14 Doing the Analysis

As mentioned in above, writing output files can be a large portion of total run time, especially on platforms with small disk write caches. Generally speaking, therefore, except during debugging, it is better to use specific REQUESTs for plotting and the graphics file for animation than to postprocess from the results (*.res*) file.

The ADAMS/Solver output (*.out*) file also is normally superfluous, since the information there is a reformatted version of the request file. You can reduce the size of the *.out* file to a minimum from the main WT menu by selecting the panel for General Output Controls and setting:

print = off

You do not need to change any of the other fields in this panel. Nothing appears in your A/View model when this panel is executed; the ADAMS dataset will have a line in it which reads, "OUTPUT/NOPRINT."

In the past, it has often been necessary to "tweak" the static and dynamic solution parameters a little to get a wind turbine model to run the most smoothly. The integrators in ADAMS/Solver version 9.1, however, have been under continuous development for some time, as have the AeroDyn aerodynamics subroutines, and you should find that the combination in WT 2.0 works much better with default integrator and solver values than previously.

In addition, the numerical performance of the AeroDyn subroutines has been significantly improved. Because of this, it is now possible to run most WT-created models using the standard, default GSTIFF integrator. We usually recommend WSTIFF, however, because it appears to run more smoothly. Despite this appearance, GSTIFF may still run much faster. You should experiment with the integrators with your model.

To specifically select WSTIFF, from the Command Navigator, select the panel for EXECUTIVE_CONTROL SET NUMERICAL_INTEGRATOR_PARAMETERS. Cycle the Integrator Type to WSTIFF and execute the panel without changing any of the other parameters. This will put the INTEGRATOR/WSTIFF command right in the dataset and you should not need to put it in your Solver command file.

Using GSTIFF instead of the BDF integrators could make your simulations run significantly faster, possibly as much as 3-4 times faster. You may, however, usually see many more informational messages about integrator and solver performance "hiccups" than previously appeared using WSTIFF. (By default, the WSTIFF integrator neglects to tell the user about most of these little problems!) As long as the integrator continues and does not get bogged down, these messages can be considered just educational. To specifically select GSTIFF, from the Command Navigator, select the panel for EXECUTIVE_CONTROL SET NUMERICAL_INTEGRATOR_PARAMETERS. Cycle the Integrator Type to GSTIFF and execute the panel without changing any of the other parameters.

At this point, your model is complete and you should change its name from the default *hawt* to *case*_1 and then write it out in dataset (.adm) format for simulation and in View binary and command file formats for safekeeping. These actions may be accomplished from the BUILD / MODEL / RENAME and FILE / EXPORT panels, or directly from the A/View command line. In the command line window, you can type:

```
model modify model=hawt new=case_1
file adams write file=case_1
file command write file=case_1 entity=case_1
file binary write file=case_1
```

Now you are ready to try your first simulation of the *case_1* turbine. Before you can do this, however, you must create a user-executable version of ADAMS/Solver that includes the AeroDyn routines. This can be done through the ADAMS graphical front end or with the menu interface from the system command line.

First, you should compile the AeroDyn FORTRAN code into object modules. In the *nrel/fortran* directory you should compile the files *aerosubs.f, modules.f, sensub.f* and g*fosub.f.* If you plan to later use the pre-configured AeroDyn requests instead of the standard ADAMS/WT requests, you can also include the *reqsub.f* file into the executable. (It is not needed for this example, but does not hurt anything to include the file.) The compile command, depending on your platform, should be something like

```
f77 -c aerosubs.f sensub.f gfosub.f modules.f     (UNIX)
                     or
df /c /G5 /Ob2 /MD aerosubs.f modules.f sensub.f gfosub.f     (NT)
```

Assuming that there were no errors in the compilation, you should end up with four object files. Note that for successful compilation, you must have the two include files *aerodyn.inc* and *bedoes.inc* in the same directory. You can then create the user-executable version of ADAMS/Solver with the menu interface step by step, or with the single long command

```
mdi -c cr-u i n aerosubs.o modules.o sensub.o gfosub.o -n wt20.exe exit
```

or for NT

```
mdi cr-u n aerosubs.obj modules.obj sensub.obj gfosub.obj -n wt20.exe
```

This should leave the file *wt20.exe* in the directory. You should copy, link or move this file into the *examples/case_1* directory for use with this model. You should then switch back to the *examples/case_1* and you will be ready to try out the *case_1* model.

Because you are running a user-executable version of ADAMS/Solver and will need special Solver commands to run it, and because you will often be running many simulations in a row, it is usually more convenient to run the code from the system command line instead of submitting it directly from the ADAMS/View.

To do this you must first create an ADAMS/Solver command file *(.acf)* to control the simulation. Using your editor, create a text file named *case_1.acf* with the following contents:

```
case_1
case_1
integrator/wstiff, err=5e-4
sim/dyn,end=0.2,step=100
sim/dyn,end=1,step=100
sim/dyn,end=2,step=53
sim/dyn,end=3,step=53
sim/dyn,end=4,step=53
sim/dyn,end=5,step=53
sim/dyn,end=6,step=53
sim/dyn,end=7,step=53
sim/dyn,end=8,step=53
sim/dyn,end=9,step=53
sim/dyn,end=10,step=53
stop
```

To run the code you can again use the menu interface step by step, or enter the single long command at the system prompt:

mdi -c ru-u i wt20.exe case_1.acf exit

or for NT

mdi ru-u wt20.exe case_1.acf

At this point, ADAMS/Solver should start up and the simulation progress should be displayed on screen. You can expect some difficulty with simulation startup, and perhaps some warning messages about corrector convergence during the run, but these can both be ignored as long as the simulation recovers. The program log is also written to the file *case_1.msg*. When the run is complete, you should be returned to the system prompt and the simulation results should be in the files *case_1.gra* and *case_1.req*. The *.msg* file should contain something very similar to the log shown below. (This is reproduced in its entirety only for the first example case:

```
   **************************************************************
   *                                                            *
   *                 Mechanical Dynamics, Inc.                  *
   *                                                            *
   *                        A D A M S                           *
   *                                                            *
   *      Automatic Dynamic Analysis of Mechanical Systems      *
   *                                                            *
   *                       Version 9.1                          *
   *                                                            *
   *  ADAMS/Solver, ADAMS/Android, ADAMS/Animation, ADAMS/FEA,  *
   *  ADAMS/Real-Time Kinematics, ADAMS/Vehicle, ADAMS/View,    *
   *      Collectively known as the ADAMS Product Line          *
   *                      copyright C 1997                      *
   *  By Mechanical Dynamics, Inc., Ann Arbor, Michigan U.S.A.  *
   *                                                            *
   **************************************************************
   *                                                            *
   *        Confidential and proprietary information of         *
   *      Mechanical Dynamics, Inc.,  Ann Arbor, Michigan       *
   *                                                            *
   *    All  rights  reserved.  This code may not be copied     *
   *    or reproduced  in any  form,  in part  or in whole,     *
   *       without the explicit prior written permission        *
   *                  of the copyright owner.                   *
   *                                                            *
   **************************************************************
   *                                                            *
   *      All product names in the ADAMS Product Line are       *
   *          trademarks of Mechanical Dynamics, Inc.           *
   *                                                            *
   **************************************************************
   *                                                            *
   *                 RESTRICTED RIGHTS LEGEND                   *
   *                                                            *
   *  If  the  Software  and  Documentation  are  provided  in  *
   *  connection  with a  government contract,  then  they are  *
   *  provided with  RESTRICTED RIGHTS.  Use,  duplication, or  *
   *  disclosure by the  Government is subject to restrictions  *
   *  as set forth in subparagraph (c)(1)(ii) of the Rights in *
   *  Technical   Data   and   Computer   Software  clause  at  *
   *  252.227-7013,  as amended.   Title  to all  intellectual  *
   *                 property remains with MDI.                 *
   *                                                            *
   **************************************************************
   **************************************************************
   *                                                            *
   *                  A D A M S / S o l v e r                   *
   *                                                            *
   *        18:54:23 24-DEC-98            Version 9.1           *
   *                                                            *
   *                                                            *
   **************************************************************


OUTFOP:IN_FILENM
     ADAMS model file .. case_1.adm

OUTFOP:OUT_FILES

   Default file names for output files

     Tabular output file:
     case_1.out

     Diagnostic file    :
     case_1.msg

     Message Database file    :
     case_1.mdb

     Graphics file      :
     case_1.gra

     Request file       :
     case_1.req

     Femdata file       :
     case_1.fem

     Results file       :
     case_1.res


INVIEW:READMDL
   Input Phase - Reading in Model


INVIEW:MESSAGE91
   ***********************************************************
   ADAMS/Solver dataset Title:
```

```
   ADAMS/View model name: case_1
   ***********************************************************

INVIEW:READ_MDL
   Reading of model complete.


INBASE:DATABASE
   Input Phase - Populating Solver database


INBASE:INP_DONE
   Input Phase Complete.


MEKINP:CPUTIME
   CPU time is 0.31045 seconds.

USRMES:USER
    SENSUB called with no errors
   ID = 1

USRMES:USER
   AeroDyn Version 11.0, University of Utah
   ID = 1

USRMES:USER
   AWT-26 ADAMS model using University of Utah aerodynamics routines v10.0
   ID = 4

USRMES:USER
    Dynamic inflow theory not used in the analysis
   ID = 5

USRMES:USER
   Only 1 line in wind file, steady wind conditions used
   ID = 7

USRMES:USER
   Detected system force units of Newtons
   ID = 1

VERINP:END_INPUT
   Input and Input Check Phase complete.

GTMODE:NUMB_DOFS
   The system has 41 kinematic degrees of freedom.

GLGETL:USER_CMND
   integrator/wstiff, err=5e-4

GLGETL:USER_CMND
   sim/dyn,end=0.2,step=100

DBANNR:BDF
   Begin the dynamic analysis.

   The system is modelled with DAEs.
   The VARIABLE coefficient BDF method will be used.

DBANNR:BDF_TABLE
   The operating values of the error tolerances for BDF are:

                              Default  |Recommended|Selected
                           |-----------|-----------|--------
         Integration error |           |           |
                 NTREL_ERR |  1.00E-03 |  -------  |  5.00E-04
                 NTABS_ERR |  1.00E-03 |  -------  |  5.00E-04
                           |           |           |
         Corrector error   |           |           |
                 CRREL_ERR |  1.00E-06 |  5.00E-07 |  5.00E-07
                 CRABS_ERR |  1.00E-06 |  5.00E-07 |  5.00E-07


ICCALC:DISPL
   Displacement initial condition analysis...

CODGEN:JAC_STAT

   Jacobian Matrix Statistics for the Initial Conditions
   ======================================================
   Number of equations ............ = 187
   Number of non-zero entries ...... = 1094
   Percentage of matrix non-zero ... = 3.1285
   Total space used in MD array .... = 79060


ICCALC:VELO
   Velocity initial condition analysis...

CODGEN:JAC_STAT

   Jacobian Matrix Statistics for the Initial Conditions
   ======================================================
   Number of equations ............ = 187
   Number of non-zero entries ...... = 1208
   Percentage of matrix non-zero ... = 3.4545
   Total space used in MD array .... = 78012


ICCALC:ACCEL
   Acceleration initial condition analysis...

CODGEN:JAC_STAT

   Jacobian Matrix Statistics for the Initial Conditions
   ======================================================
   Number of equations ............ = 388
   Number of non-zero entries ...... = 2398
   Percentage of matrix non-zero ... = 1.5929
   Total space used in MD array .... = 91020


SYMBLU:DISP_VELO
   Generating the Jacobian matrix for the displacements and velocities.

CODGEN:JAC_STAT

   Jacobian Matrix Statistics for the Initial Conditions
   ======================================================
   Number of equations ............ = 187
   Number of non-zero entries ...... = 1094
   Percentage of matrix non-zero ... = 3.1285
   Total space used in MD array .... = 79166


SYMBLU:ACCELRATN
   Generating the Jacobian matrix for the accelerations and forces.

CODGEN:JAC_STAT

   Jacobian Matrix Statistics for the Initial Conditions
   ======================================================
   Number of equations ............ = 388
   Number of non-zero entries ...... = 2398
   Percentage of matrix non-zero ... = 1.5929
   Total space used in MD array .... = 104674


SYMBLU:DYNAMICS
   Generating the Jacobian matrix for the dynamics problem.

CODGEN:JAC_STAT

   Jacobian Matrix Statistics for a Dynamic Analysis
   ======================================================
   Number of equations ............ = 559
   Number of non-zero entries ...... = 4982
   Percentage of matrix non-zero ... = 1.5943
   Total space used in MD array .... = 173552

    Simulation         Time         Cumulative    Cumulative     Integrator
       Time            Step         Iterations    Steps Taken      Order
    ___________      ___________    __________    ___________    __________
    0.00000E+00      1.00000E-04          0             0             1

CORRCT:NO_CONVG

 ---- WARNING ----
   The corrector has failed to converge at time 1.89063E-04 after 6 attempts.

GTMODE:NUMB_DOFS
   The system has 41 kinematic degrees of freedom.

WSTIFF:RESTART
   At time 1.0E-04 and step 2 , ADAMS is adjusting solver and integrator
   parameters to attempt an integration restart.

ICCALC:DISPL
   Displacement initial condition analysis...

CODGEN:JAC_STAT

   Jacobian Matrix Statistics for the Initial Conditions
   ======================================================
   Number of equations ............ = 187
   Number of non-zero entries ...... = 1094
   Percentage of matrix non-zero ... = 3.1285
   Total space used in MD array .... = 78892


ICCALC:VELO
   Velocity initial condition analysis...

CODGEN:JAC_STAT
```

```
   Jacobian Matrix Statistics for the Initial Conditions
   ======================================================
   Number of equations ............ = 187
   Number of non-zero entries ...... = 1094
   Percentage of matrix non-zero ... = 3.1285
   Total space used in MD array .... = 78892


ICCALC:ACCEL
   Acceleration initial condition analysis...

CODGEN:JAC_STAT

   Jacobian Matrix Statistics for the Initial Conditions
   ======================================================
   Number of equations ............ = 388
   Number of non-zero entries ...... = 2398
   Percentage of matrix non-zero ... = 1.5929
   Total space used in MD array .... = 92046


SYMBLU:DISP_VELO
   Generating the Jacobian matrix for the displacements and velocities.

CODGEN:JAC_STAT

   Jacobian Matrix Statistics for the Initial Conditions
   ======================================================
   Number of equations ............ = 187
   Number of non-zero entries ...... = 1094
   Percentage of matrix non-zero ... = 3.1285
   Total space used in MD array .... = 78976


SYMBLU:ACCELRATN
   Generating the Jacobian matrix for the accelerations and forces.

CODGEN:JAC_STAT

   Jacobian Matrix Statistics for the Initial Conditions
   ======================================================
   Number of equations ............ = 388
   Number of non-zero entries ...... = 2398
   Percentage of matrix non-zero ... = 1.5929
   Total space used in MD array .... = 105508


SYMBLU:DYNAMICS
   Generating the Jacobian matrix for the dynamics problem.

CODGEN:JAC_STAT

   Jacobian Matrix Statistics for a Dynamic Analysis
   ======================================================
   Number of equations ............ = 559
   Number of non-zero entries ...... = 4982
   Percentage of matrix non-zero ... = 1.5943
   Total space used in MD array .... = 179208

    Simulation        Time        Cumulative    Cumulative     Integrator
       Time           Step        Iterations    Steps Taken      Order
    ___________    ___________    __________    ___________    __________
    1.00000E-04    1.00000E-04        69              1            1
    2.00000E-03    1.20729E-03        78              4             1
    2.00000E-02    2.00000E-03       105             13             2
    4.00000E-02    2.00000E-03       135             23             2
    6.00000E-02    2.00000E-03       158             33             2
    8.00000E-02    2.00000E-03       178             43             2
    1.00000E-01    2.00000E-03       198             53             2
    1.20000E-01    2.00000E-03       219             63             2
    1.40000E-01    2.00000E-03       239             73             2
    1.60000E-01    2.00000E-03       259             83             2
    1.80000E-01    2.00000E-03       279             93             2
    2.00000E-01    2.00000E-03       299            103             2

GLGETL:USER_CMND
   sim/dyn,end=1,step=100

DBANNR:BDF
   Begin the dynamic analysis.

   The system is modelled with DAEs.
   The VARIABLE coefficient BDF method will be used.

DBANNR:BDF_TABLE
   The operating values of the error tolerances for BDF are:

                          Default   |Recommended|Selected
                       |-----------|-----------|--------
        Integration error |           |           |
                NTREL_ERR |  1.00E-03 |  -------  |  5.00E-04
                NTABS_ERR |  1.00E-03 |  -------  |  5.00E-04
                          |           |           |
         Corrector error   |           |           |
                 CRREL_ERR |  1.00E-06 |  5.00E-07 |  5.00E-07
                 CRABS_ERR |  1.00E-06 |  5.00E-07 |  5.00E-07

    Simulation         Time         Cumulative   Cumulative      Integrator
       Time            Step         Iterations    Steps Taken       Order
    ___________      ___________    __________    ___________    __________
    2.00000E-01      4.00000E-04         299            103           2
    2.08000E-01      6.00000E-03         305            105           2
    2.80000E-01      8.00000E-03         333            114           2
    3.60000E-01      8.00000E-03         362            124           4
    4.40000E-01      8.00000E-03         392            134           3
    5.20000E-01      8.00000E-03         421            144           3
    6.00000E-01      8.00000E-03         450            154           2
    6.80000E-01      8.00000E-03         474            164           4
    7.60000E-01      8.00000E-03         494            174           3
    8.40000E-01      8.00000E-03         514            184           3
    9.20000E-01      8.00000E-03         534            194           3
    1.00000E+00      8.00000E-03         554            204           3

GLGETL:USER_CMND
   sim/dyn,end=2,step=53

DBANNR:BDF
   Begin the dynamic analysis.

   The system is modelled with DAEs.
   The VARIABLE coefficient BDF method will be used.

DBANNR:BDF_TABLE
   The operating values of the error tolerances for BDF are:

                              Default  |Recommended|Selected
                           |-----------|-----------|--------
         Integration error |           |           |
                 NTREL_ERR |  1.00E-03 |  -------  |  5.00E-04
                 NTABS_ERR |  1.00E-03 |  -------  |  5.00E-04
                           |           |           |
         Corrector error   |           |           |
                 CRREL_ERR |  1.00E-06 |  5.00E-07 |  5.00E-07
                 CRABS_ERR |  1.00E-06 |  5.00E-07 |  5.00E-07

    Simulation         Time         Cumulative   Cumulative      Integrator
       Time            Step         Iterations    Steps Taken       Order
    ___________      ___________    __________    ___________    __________
    1.00000E+00      9.43396E-04         554            204           3
    1.01887E+00      1.08679E-02         559            206           3
    1.11321E+00      1.88679E-02         574            211           3
    1.22642E+00      1.88679E-02         592            217           4
    1.33962E+00      1.88679E-02         610            223           4
    1.45283E+00      1.88679E-02         628            229           3
    1.56604E+00      1.88679E-02         646            235           4
    1.67925E+00      1.88679E-02         664            241           5
    1.79245E+00      1.88679E-02         682            247           5
    1.90566E+00      1.88679E-02         700            253           5

GLGETL:USER_CMND
   sim/dyn,end=3,step=53

DBANNR:BDF
   Begin the dynamic analysis.

   The system is modelled with DAEs.
   The VARIABLE coefficient BDF method will be used.

DBANNR:BDF_TABLE
   The operating values of the error tolerances for BDF are:

                              Default  |Recommended|Selected
                           |-----------|-----------|--------
         Integration error |           |           |
                 NTREL_ERR |  1.00E-03 |  -------  |  5.00E-04
                 NTABS_ERR |  1.00E-03 |  -------  |  5.00E-04
                           |           |           |
         Corrector error   |           |           |
                 CRREL_ERR |  1.00E-06 |  5.00E-07 |  5.00E-07
                 CRABS_ERR |  1.00E-06 |  5.00E-07 |  5.00E-07

    Simulation         Time         Cumulative   Cumulative      Integrator
       Time            Step         Iterations    Steps Taken       Order
    ___________      ___________    __________    ___________    __________
    2.00000E+00      9.43396E-04         715            258           4
    2.01887E+00      1.88679E-02         718            259           5
    2.11321E+00      7.07547E-03         772            270           2
    2.22642E+00      1.41509E-02         843            284           2
    2.33962E+00      9.43396E-03         957            304           2
    2.45283E+00      1.88679E-02        1066            322           3
    2.56604E+00      1.88679E-02        1097            330           3
    2.67925E+00      1.88679E-02        1115            336           3
    2.79245E+00      1.88679E-02        1133            342           3
    2.90566E+00      1.41509E-02        1172            351           2

GLGETL:USER_CMND
```

```
   sim/dyn,end=4,step=53

DBANNR:BDF
   Begin the dynamic analysis.

   The system is modelled with DAEs.
   The VARIABLE coefficient BDF method will be used.

DBANNR:BDF_TABLE
   The operating values of the error tolerances for BDF are:

                              Default  |Recommended|Selected
                           |-----------|-----------|--------
         Integration error |           |           |
                 NTREL_ERR |  1.00E-03 |  -------  |  5.00E-04
                 NTABS_ERR |  1.00E-03 |  -------  |  5.00E-04
                           |           |           |
         Corrector error   |           |           |
                 CRREL_ERR |  1.00E-06 |  5.00E-07 |  5.00E-07
                 CRABS_ERR |  1.00E-06 |  5.00E-07 |  5.00E-07

    Simulation         Time         Cumulative   Cumulative      Integrator
       Time            Step         Iterations    Steps Taken      Order
    ___________      ___________    __________    ___________    __________
    3.00000E+00      9.43396E-04        1227            361            3
    3.01887E+00      1.88679E-02        1231            362            2
    3.11321E+00      9.43396E-03        1284            373            3
    3.22642E+00      1.41509E-02        1336            384            3
    3.33962E+00      1.88679E-02        1381            396            3
    3.45283E+00      1.88679E-02        1445            409            3
    3.56604E+00      1.88679E-02        1469            415            5
    3.67925E+00      1.88679E-02        1509            423            5
    3.79245E+00      1.88679E-02        1535            429            5
    3.90566E+00      9.43396E-03        1565            438            3

GLGETL:USER_CMND
   sim/dyn,end=5,step=53

DBANNR:BDF
   Begin the dynamic analysis.

   The system is modelled with DAEs.
   The VARIABLE coefficient BDF method will be used.

DBANNR:BDF_TABLE
   The operating values of the error tolerances for BDF are:

                              Default  |Recommended|Selected
                           |-----------|-----------|--------
         Integration error |           |           |
                 NTREL_ERR |  1.00E-03 |  -------  |  5.00E-04
                 NTABS_ERR |  1.00E-03 |  -------  |  5.00E-04
                           |           |           |
         Corrector error   |           |           |
                 CRREL_ERR |  1.00E-06 |  5.00E-07 |  5.00E-07
                 CRABS_ERR |  1.00E-06 |  5.00E-07 |  5.00E-07

    Simulation         Time         Cumulative   Cumulative      Integrator
       Time            Step         Iterations    Steps Taken      Order
    ___________      ___________    __________    ___________    __________
    4.00000E+00      9.43396E-04        1606            447            4
    4.01887E+00      1.88679E-02        1610            448            4
    4.11321E+00      1.88679E-02        1644            455            3
    4.22642E+00      1.88679E-02        1669            461            5
    4.33962E+00      4.71698E-03        1713            472            3
    4.45283E+00      7.22288E-03        1785            487            3
    4.56604E+00      1.88679E-02        1852            500            3
    4.67925E+00      1.88679E-02        1879            506            4
    4.79245E+00      1.88679E-02        1903            512            5
    4.90566E+00      1.88679E-02        1946            523            3

GLGETL:USER_CMND
   sim/dyn,end=6,step=53

DBANNR:BDF
   Begin the dynamic analysis.

   The system is modelled with DAEs.
   The VARIABLE coefficient BDF method will be used.

DBANNR:BDF_TABLE
   The operating values of the error tolerances for BDF are:

                              Default  |Recommended|Selected
                           |-----------|-----------|--------
         Integration error |           |           |
                 NTREL_ERR |  1.00E-03 |  -------  |  5.00E-04
                 NTABS_ERR |  1.00E-03 |  -------  |  5.00E-04
                           |           |           |
         Corrector error   |           |           |
                 CRREL_ERR |  1.00E-06 |  5.00E-07 |  5.00E-07
                 CRABS_ERR |  1.00E-06 |  5.00E-07 |  5.00E-07
    Simulation         Time         Cumulative   Cumulative     Integrator
       Time            Step         Iterations    Steps Taken      Order
    ___________      ___________    __________    ___________    __________
    5.00000E+00      9.43396E-04        1963            528           4
    5.01887E+00      1.88679E-02        1967            529           4
    5.11321E+00      1.41509E-02        2012            539           3
    5.22642E+00      1.88679E-02        2037            545           4
    5.33962E+00      1.88679E-02        2061            551           5
    5.45283E+00      1.88679E-02        2103            561           3
    5.56604E+00      1.88679E-02        2149            571           4
    5.67925E+00      1.88679E-02        2185            579           3
    5.79245E+00      1.88679E-02        2207            585           5
    5.90566E+00      1.88679E-02        2248            596           3

GLGETL:USER_CMND
   sim/dyn,end=7,step=53

DBANNR:BDF
   Begin the dynamic analysis.

   The system is modelled with DAEs.
   The VARIABLE coefficient BDF method will be used.

DBANNR:BDF_TABLE
   The operating values of the error tolerances for BDF are:

                              Default  |Recommended|Selected
                           |-----------|-----------|--------
         Integration error |           |           |
                 NTREL_ERR |  1.00E-03 |  -------  |  5.00E-04
                 NTABS_ERR |  1.00E-03 |  -------  |  5.00E-04
                           |           |           |
         Corrector error   |           |           |
                 CRREL_ERR |  1.00E-06 |  5.00E-07 |  5.00E-07
                 CRABS_ERR |  1.00E-06 |  5.00E-07 |  5.00E-07

    Simulation         Time         Cumulative   Cumulative     Integrator
       Time            Step         Iterations    Steps Taken      Order
    ___________      ___________    __________    ___________    __________
    6.00000E+00      9.43396E-04        2284            604           4
    6.01887E+00      1.88679E-02        2288            605           4
    6.11321E+00      1.88679E-02        2326            614           4
    6.22642E+00      1.88679E-02        2360            622           4
    6.33962E+00      1.88679E-02        2384            628           4
    6.45283E+00      1.88679E-02        2421            638           4
    6.56604E+00      9.43396E-03        2459            647           2
    6.67925E+00      1.41509E-02        2508            656           3
    6.79245E+00      1.88679E-02        2548            665           4
    6.90566E+00      9.43396E-03        2612            679           3

GLGETL:USER_CMND
   sim/dyn,end=8,step=53

DBANNR:BDF
   Begin the dynamic analysis.

   The system is modelled with DAEs.
   The VARIABLE coefficient BDF method will be used.

DBANNR:BDF_TABLE
   The operating values of the error tolerances for BDF are:

                              Default  |Recommended|Selected
                           |-----------|-----------|--------
         Integration error |           |           |
                 NTREL_ERR |  1.00E-03 |  -------  |  5.00E-04
                 NTABS_ERR |  1.00E-03 |  -------  |  5.00E-04
                           |           |           |
         Corrector error   |           |           |
                 CRREL_ERR |  1.00E-06 |  5.00E-07 |  5.00E-07
                 CRABS_ERR |  1.00E-06 |  5.00E-07 |  5.00E-07

    Simulation         Time         Cumulative   Cumulative     Integrator
       Time            Step         Iterations    Steps Taken      Order
    ___________      ___________    __________    ___________    __________
    7.00000E+00      9.43396E-04        2630            685           5
    7.01887E+00      1.88679E-02        2633            686           5
    7.11321E+00      1.88679E-02        2651            692           4
    7.22642E+00      1.88679E-02        2669            698           6
    7.33962E+00      1.88679E-02        2689            704           4
    7.45283E+00      1.88679E-02        2716            713           3
    7.56604E+00      9.43396E-03        2746            721           2
    7.67925E+00      1.88679E-02        2775            727           4
    7.79245E+00      1.88679E-02        2808            735           4
    7.90566E+00      1.88679E-02        2841            744           4

GLGETL:USER_CMND
   sim/dyn,end=9,step=53

DBANNR:BDF
   Begin the dynamic analysis.

   The system is modelled with DAEs.
   The VARIABLE coefficient BDF method will be used.

DBANNR:BDF_TABLE
   The operating values of the error tolerances for BDF are:

                              Default  |Recommended|Selected
                           |-----------|-----------|--------
         Integration error |           |           |
                 NTREL_ERR |  1.00E-03 |  -------  |  5.00E-04
                 NTABS_ERR |  1.00E-03 |  -------  |  5.00E-04
                           |           |           |
         Corrector error |             |           |
                 CRREL_ERR |  1.00E-06 |  5.00E-07 |  5.00E-07
                 CRABS_ERR |  1.00E-06 |  5.00E-07 |  5.00E-07

    Simulation         Time         Cumulative   Cumulative     Integrator
       Time            Step         Iterations    Steps Taken      Order
    ___________      ___________    __________    ___________    __________
    8.00000E+00      9.43396E-04        2861            749           4
    8.01887E+00      1.88679E-02        2865            750           4
    8.11321E+00      1.88679E-02        2881            755           4
    8.22642E+00      1.88679E-02        2899            761           5
    8.33962E+00      9.43396E-03        2925            768           4
    8.45283E+00      1.88679E-02        2955            777           4
    8.56604E+00      1.88679E-02        2988            785           3
    8.67925E+00      1.88679E-02        3015            791           4
    8.79245E+00      1.88679E-02        3037            797           6
    8.90566E+00      9.43396E-03        3073            807           3

GLGETL:USER_CMND
   sim/dyn,end=10,step=53

DBANNR:BDF
   Begin the dynamic analysis.

   The system is modelled with DAEs.
   The VARIABLE coefficient BDF method will be used.

DBANNR:BDF_TABLE
   The operating values of the error tolerances for BDF are:

                              Default  |Recommended|Selected
                           |-----------|-----------|--------
         Integration error |           |           |
                 NTREL_ERR |  1.00E-03 |  -------  |  5.00E-04
                 NTABS_ERR |  1.00E-03 |  -------  |  5.00E-04
                           |           |           |
         Corrector error |             |           |
                 CRREL_ERR |  1.00E-06 |  5.00E-07 |  5.00E-07
                 CRABS_ERR |  1.00E-06 |  5.00E-07 |  5.00E-07

    Simulation         Time         Cumulative   Cumulative     Integrator
       Time            Step         Iterations    Steps Taken      Order
    ___________      ___________    __________    ___________    __________
    9.00000E+00      9.43396E-04        3092            812           4
    9.01887E+00      1.88679E-02        3096            813           4
    9.11321E+00      1.88679E-02        3117            820           4
    9.22642E+00      1.88679E-02        3135            826           5
    9.33962E+00      6.28931E-03        3164            834           3
    9.45283E+00      1.88679E-02        3192            843           4
    9.56604E+00      1.88679E-02        3211            849           4
    9.67925E+00      1.88679E-02        3229            855           5
    9.79245E+00      1.88679E-02        3247            861           5
    9.90566E+00      1.88679E-02        3282            871           3

GLGETL:USER_CMND
   stop

TERM0:EXE_TERM

      ADAMS/Solver execution terminated by subprogram A3TERM


TERM0:CP_TIME
   CPU time used = 81.868 seconds
```

7.15 Static Solution Note

Getting a valid static equilibrium solution is surprisingly difficult for many rotor models, including this example case. If you really need an equilibrium solution, you should first try with the default EQUILIBRIUM statement parameters. If this fails, you should try reducing the alimit parameter on the EQUILIBRIUM statement to about 5 degrees and the tlimit value to about 100 meters, while also increasing the stability parameter to about .01 and increasing the maxit value to 100. After you get a solution, you can "play" these parameters to get optimal convergence. Note that with no line voltage and no wind at startup, you may get either horizontal or vertical blade static solutions for a 2-bladed rotor, even though a vertical solution probably has slightly less total energy. You might also get solutions with the blades bent into unusual positions. Do not use these as a starting point for a dynamic simulation!

7.16 Visualizing the Results

At this point, you are ready to read the results of the simulation back into ADAMS/View to look at the responses. Switch back to the A/View window and either use the FILE IMPORT menu or enter at the View command line:

file analysis read file=case_1 model=case_1

It will take View a few moments to read in the data from the graphics (*case_1.gra*) and request (*case_1.req*) files. You can then animate the results and see how the rotor responded. There are quite a few ways to animate response in View. The simplest way in the WT interface is to bring up the control panel and just hit the ANIMATE button.

7.17 Plotting Output

ADAMS/View 9.1 has a completely new plotting interface, including a large number of plotting features which can be accessed in many ways. Quick plots of request data can be made by easily made using the Plot Builder. The data can also be "surfed" this way.

For repetitive plotting of specific requests from multiple simulations, it is often best to create a View command file (*.cmd*) containing the necessary commands to create and customize all the plots for a particular run. This command file contains the same commands you could execute via the plot builder or type in at the View command line to create the plots, but is easily modifiable using a text editor for customization and changes. An example of such a command file is found in the file *plotemup.cmd* in the *examples/case_1* directory. The contents are repeated here:

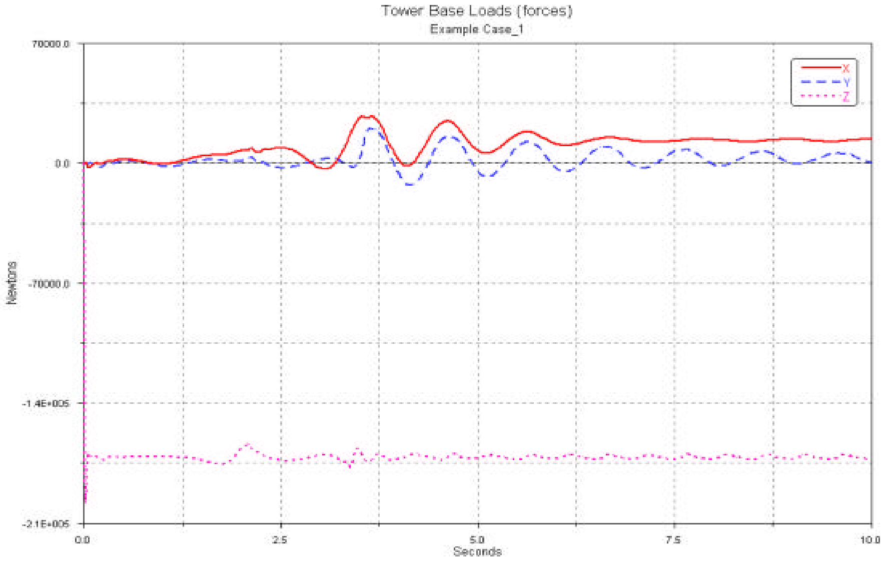
```
! View command file to plot results from example case 1
! Created by A. Elliott, MDI, December '98.

xy_plot template create plot=tbl1 title="Tower Base Loads (forces)"&
 subtitle="Example Case_1" vlabel="Newtons" hlabel="Seconds" &
 auto=no
xy_plot curve create plot=tbl1 legend=yes &
 vaxis=twr_base_loads/X,twr_base_loads/Y,twr_base_loads/Z
xy att plot_name = .tbl1 graph_area = 20, 5, 140, 85
xy curve mod curve=.tbl1.curve   legend="X"
xy curve mod curve=.tbl1.curve_2 legend="Y"
xy curve mod curve=.tbl1.curve_3 legend="Z"
note att note=.tbl1.curve.legend   point_size = 8
note att note=.tbl1.curve_2.legend point_size = 8
note att note=.tbl1.curve_3.legend point_size = 8

xy_plot template create plot=tbl2 title="Tower Base Loads (torques)"&
 subtitle="Example Case_1" vlabel="Newton-Meters" hlabel="Seconds" &
 auto=no
xy_plot curve create plot=tbl2 legend=yes &
 vaxis=twr_base_loads/R1,twr_base_loads/R2,twr_base_loads/R3
xy att plot_name = .tbl2 graph_area = 20, 5, 140, 85
xy curve mod curve=.tbl2.curve   legend="X"
xy curve mod curve=.tbl2.curve_2 legend="Y"
xy curve mod curve=.tbl2.curve_3 legend="Z"
note att note=.tbl2.curve.legend   point_size = 8
note att note=.tbl2.curve_2.legend  point_size = 8
note att note=.tbl2.curve_3.legend  point_size = 8

xy_plot template create plot=yaw1 title="Yaw Table Loads (forces)"&
 subtitle="Example Case_1" vlabel="Newtons" hlabel="Seconds" &
 auto=no
xy_plot curve create plot=yaw1 legend=yes &
 vaxis=yaw_table_loads/X,yaw_table_loads/Y,yaw_table_loads/Z
xy att plot_name = .yaw1 graph_area = 20, 5, 140, 85
xy curve mod curve=.yaw1.curve   legend="X"
xy curve mod curve=.yaw1.curve_2 legend="Y"
xy curve mod curve=.yaw1.curve_3 legend="Z"
note att note=.yaw1.curve.legend    point_size = 8
note att note=.yaw1.curve_2.legend  point_size = 8
note att note=.yaw1.curve_3.legend  point_size = 8


xy_plot template create plot=yaw2 title="Yaw Table Loads (torques)"&
 subtitle="Example Case_1" vlabel="Newton-Meters" hlabel="Seconds" &
 auto=no
xy_plot curve create plot=yaw2 legend=yes &
 vaxis=yaw_table_loads/R1,yaw_table_loads/R2,yaw_table_loads/R3
xy att plot_name = .yaw2 graph_area = 20, 5, 140, 85
xy curve mod curve=.yaw2.curve   legend="X"
xy curve mod curve=.yaw2.curve_2 legend="Y"
xy curve mod curve=.yaw2.curve_3 legend="Z"
note att note=.yaw2.curve.legend    point_size = 8
note att note=.yaw2.curve_2.legend  point_size = 8
note att note=.yaw2.curve_3.legend  point_size = 8

xy_plot template create plot=rpm  title="Rotor Speed"&
 subtitle="Example Case_1" vlabel="RPM" hlabel="Seconds" legend=no &
 auto=no
xy_plot curve create plot=rpm vaxis=rotor_data/Z
xy att plot_name = .rpm graph_area = 20, 5, 140, 85

xy_plot template create plot=yaw  title="Nacelle Yaw"&
 subtitle="Example Case_1" vlabel="Degrees" hlabel="Seconds" legend=no &
 auto=no
xy_plot curve create plot=yaw vaxis=rotor_data/R1
xy att plot_name = .yaw graph_area = 20, 5, 140, 85

xy_plot template create plot=root1 title="Blade Root Loads (forces)"&
 subtitle="Example Case_1" vlabel="Newtons" hlabel="Seconds" &
 auto=no
xy_plot curve create plot=root1 legend=yes &
 vaxis=root_loads/X,root_loads/Y,root_loads/Z
xy att plot_name = .root1 graph_area = 20, 5, 140, 85
xy curve mod curve=.root1.curve   legend="X"
xy curve mod curve=.root1.curve_2 legend="Y"
xy curve mod curve=.root1.curve_3 legend="Z"
note att note=.root1.curve.legend    point_size = 8
note att note=.root1.curve_2.legend  point_size = 8
note att note=.root1.curve_3.legend  point_size = 8

xy_plot template create plot=root2 title="Blade Root Loads (torques)"&
 subtitle="Example Case_1" vlabel="Newton-Meters" hlabel="Seconds" &
 auto=no
xy_plot curve create plot=root2 legend=yes &
 vaxis=root_loads/R1,root_loads/R2,root_loads/R3
xy att plot_name = .root2 graph_area = 20, 5, 140, 85
xy curve mod curve=.root2.curve   legend="X"
xy curve mod curve=.root2.curve_2 legend="Y"
xy curve mod curve=.root2.curve_3 legend="Z"
note att note=.root2.curve.legend    point_size = 8
note att note=.root2.curve_2.legend  point_size = 8
note att note=.root2.curve_3.legend  point_size = 8

xy_plot template create plot=teeter  title="Teetering Angle"&
 subtitle="Example Case_1" vlabel="Degrees" hlabel="Seconds" legend=no &
 auto=no
xy_plot curve create plot=teeter vaxis=teetering_hub/X
xy att plot_name = .teeter graph_area = 20, 5, 140, 85

xy_plot template create plot=azimuth title="Blade 1 Azimuth Angle"&
```

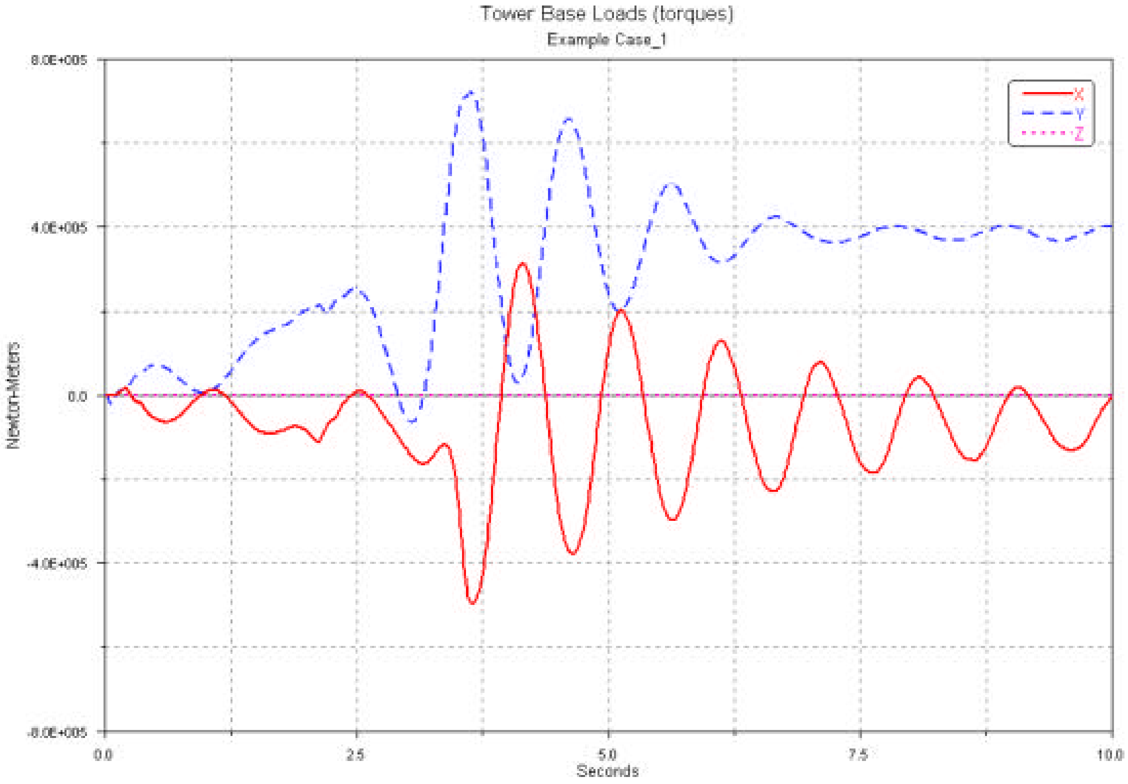
```
         subtitle="Example Case_1" vlabel="Degrees" hlabel="Seconds" legend=no &
         auto=no
        xy_plot curve create plot=azimuth vaxis=teetering_hub/Y
        xy att plot_name = .azimuth graph_area = 20, 5, 140, 85

        xy_plot template create plot=torque title="Motor-Generator Torque"&
         subtitle="Example Case_1" vlabel="Newton-Meters" hlabel="Seconds" legend=no &
         auto=no
        xy_plot curve create plot=torque vaxis=motor_generator/X
        xy att plot_name = .torque graph_area = 20, 5, 140, 85
```

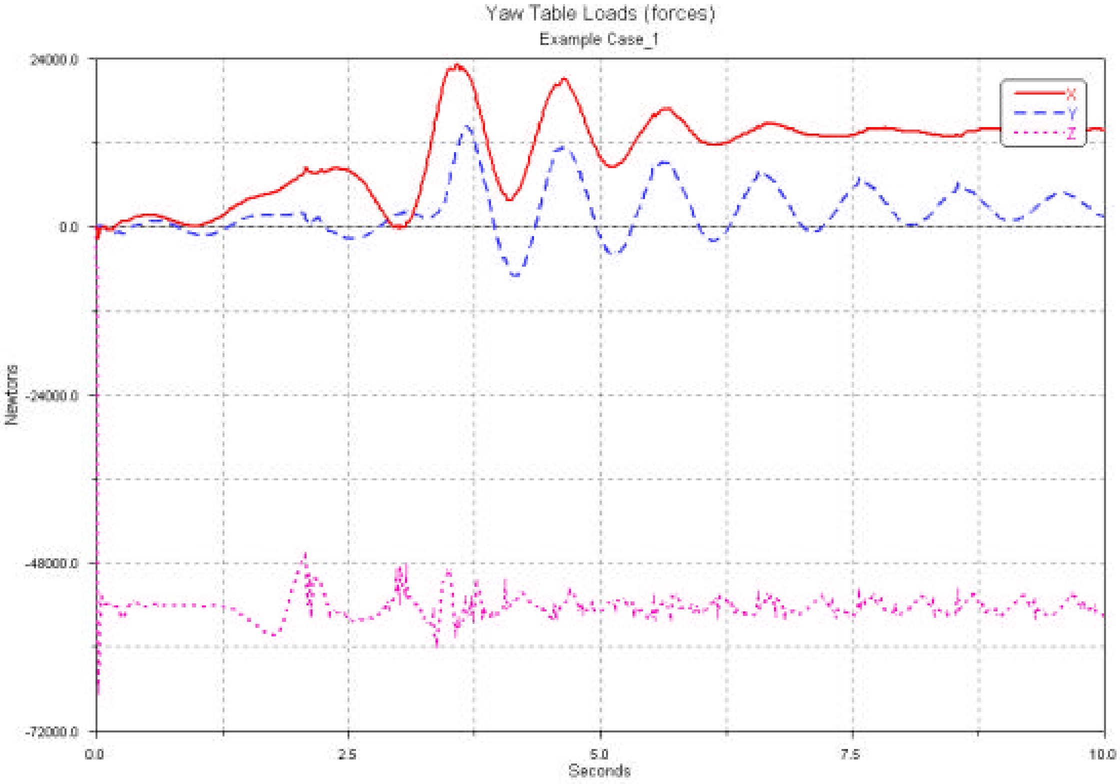
You can read in and run this command file through the FILE / IMPORT menus or by entering at the View command line:

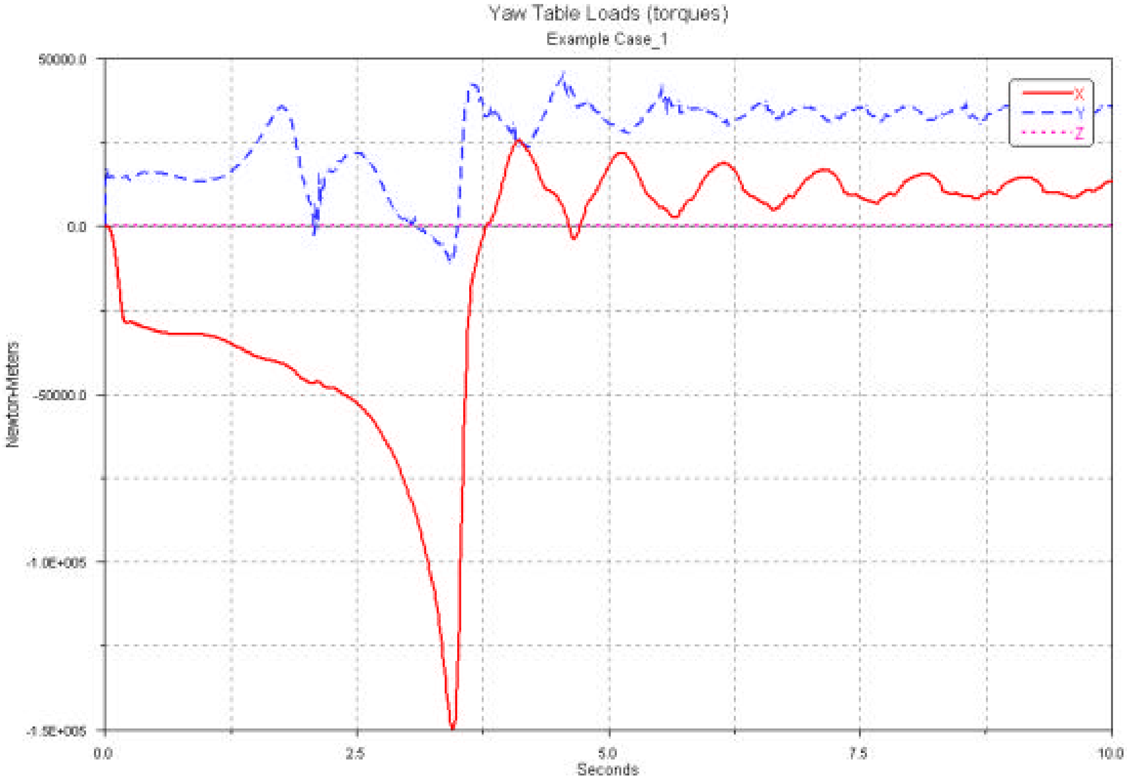
file command read file=plotemup

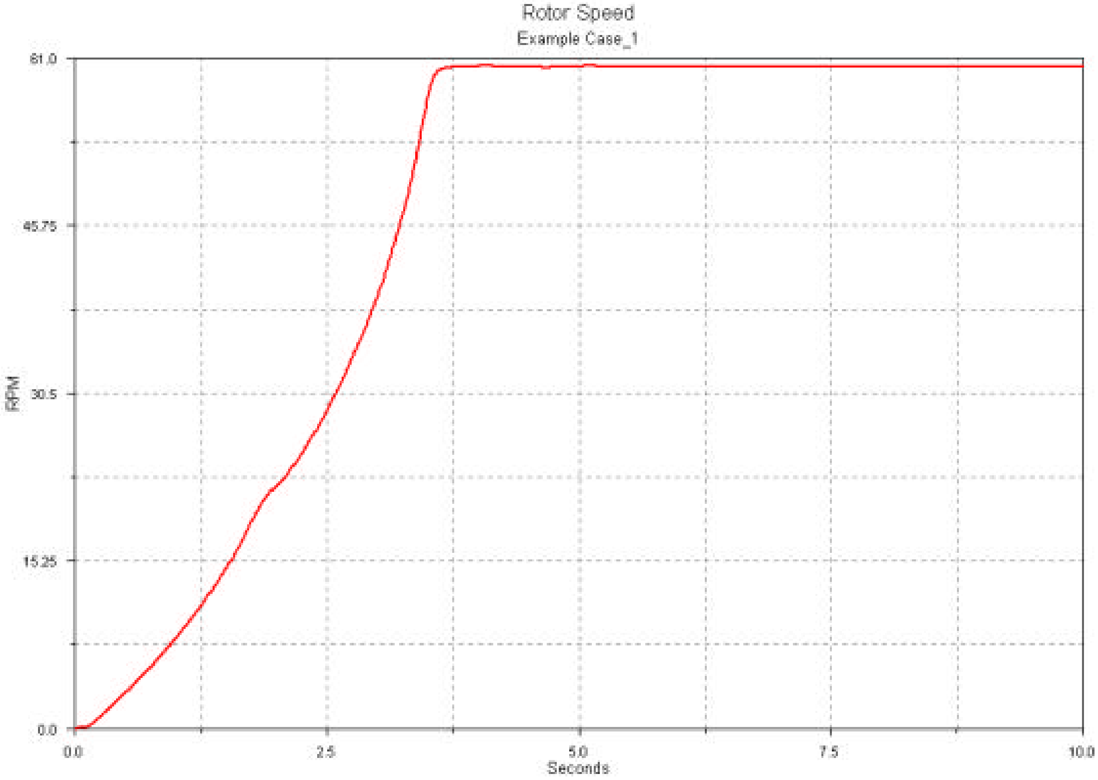
The example plots below can be used to confirm that your model and WT executable are working correctly

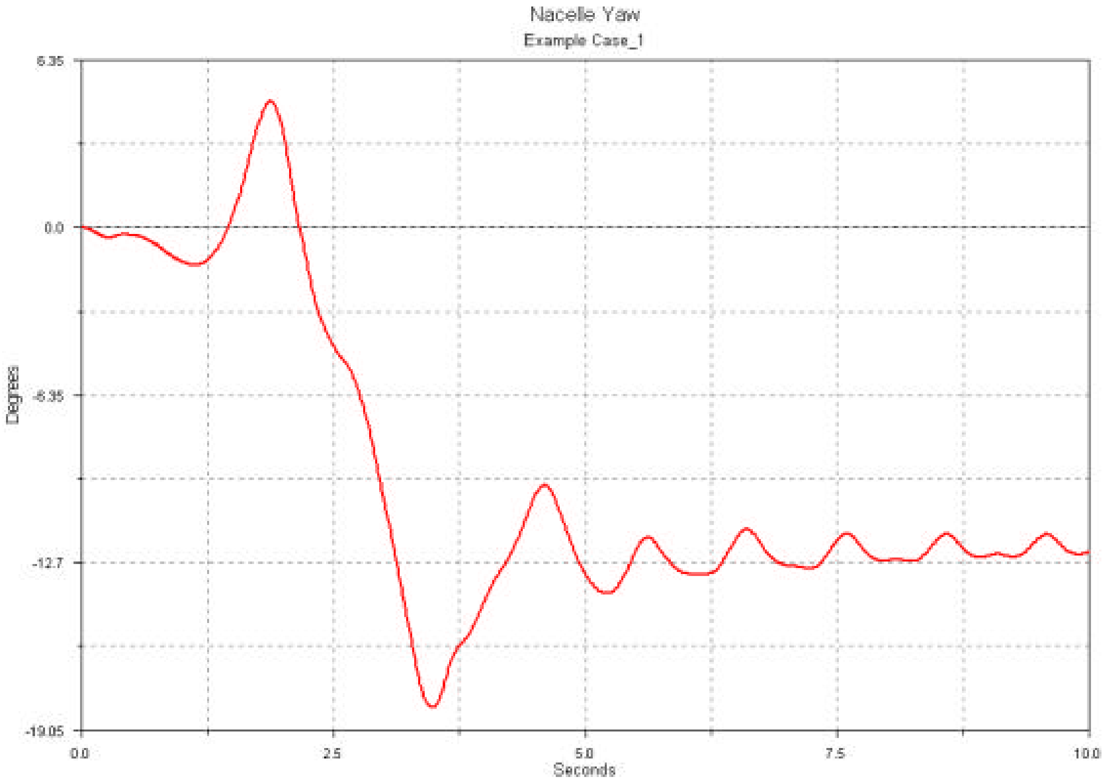


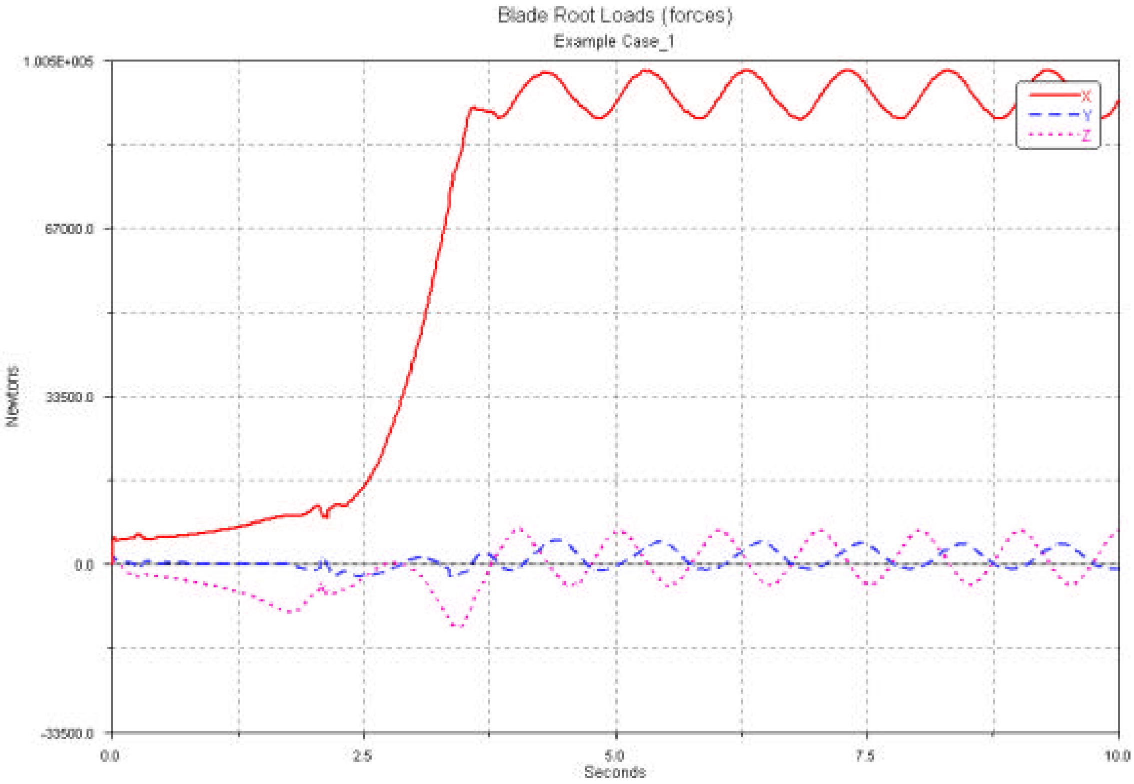

Tower Base Loads (torques)
Example Case_1
X
Y
Z
8.0E+005
4.0E+005
0.0
-4.0E+005
-8.0E+005
Newton-Meters
0.0
2.5
5.0
7.5
10.0
Seconds

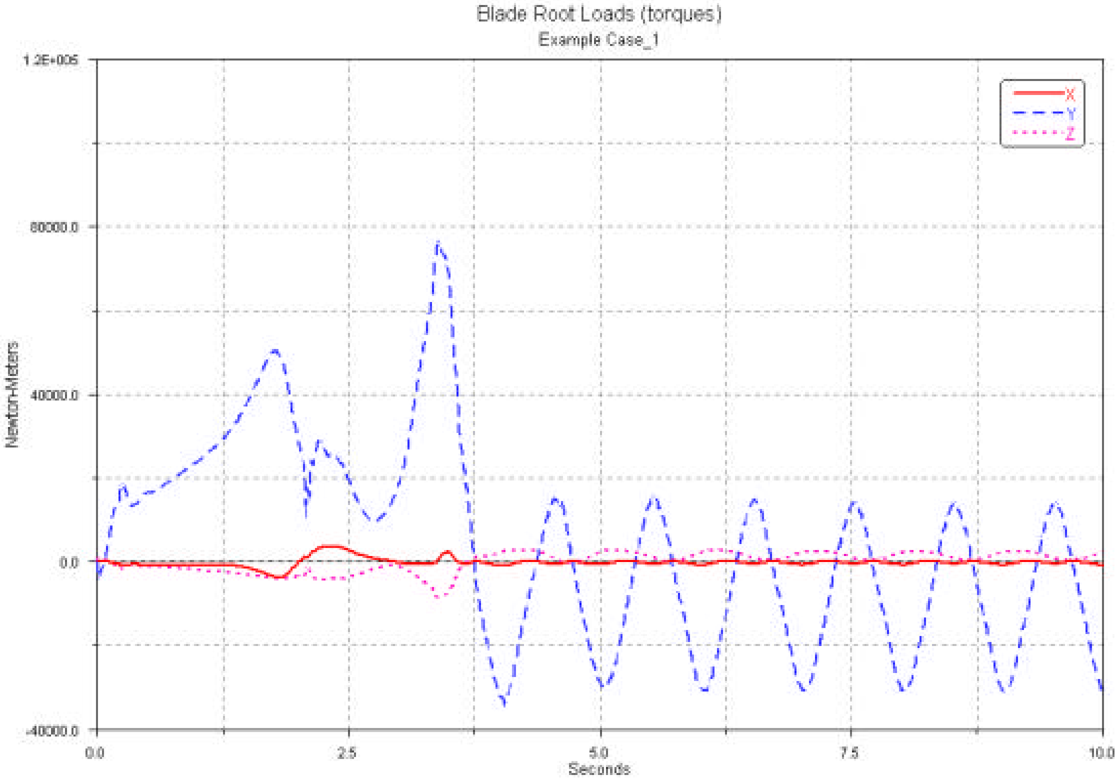

Yaw Table Loads (forces)
Example Case_1
X
Y
Z
24000.0
0.0
-24000.0
-48000.0
-72000.0
Newtons
0.0
2.5
5.0
7.5
10.0
Seconds

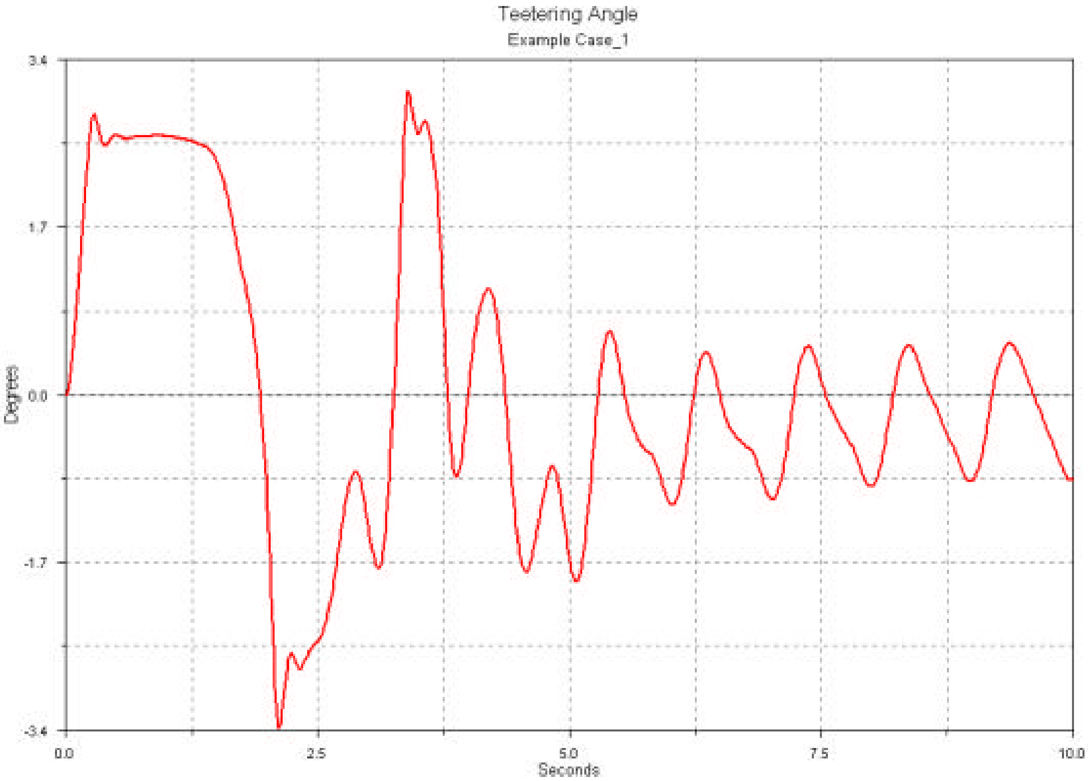

Yaw Table Loads (torques)
Example Case_1
50000.0
0.0
-50000.0
-1.0E+005
-1.5E+005
Newton-Meters
0.0
2.5
5.0
7.5
10.0
Seconds
X
Y
Z

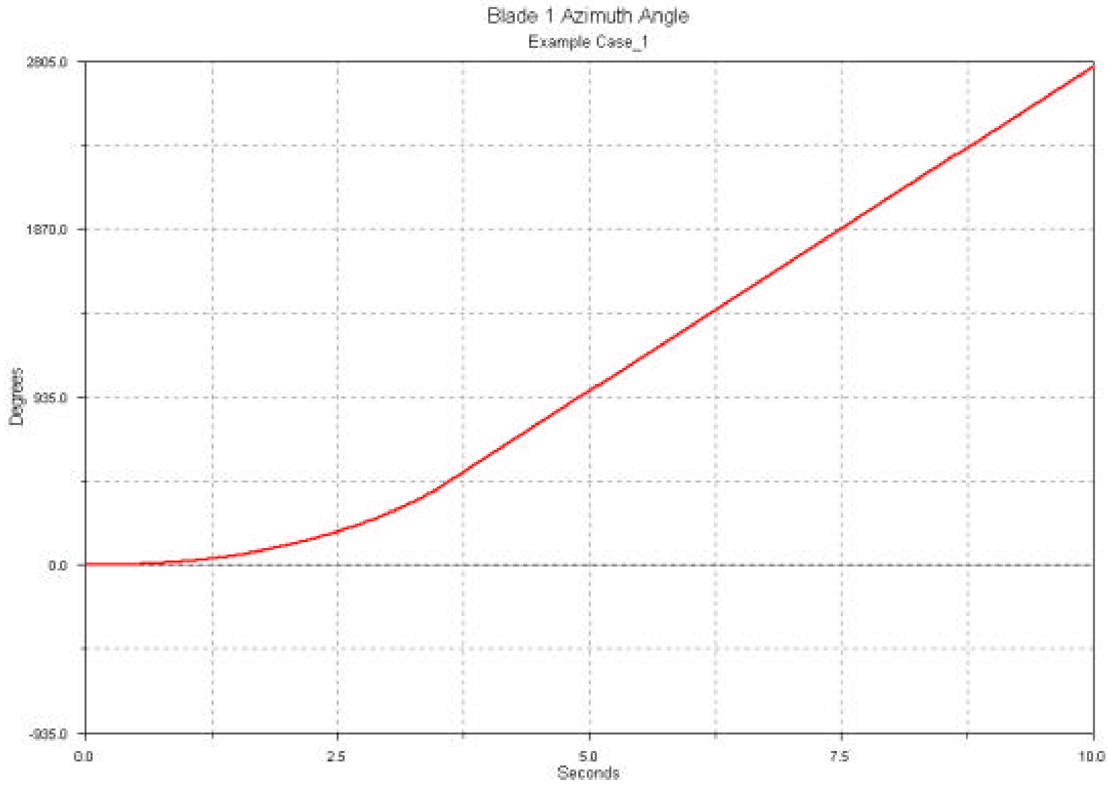

Rotor Speed
Example Case_1
61.0
45.75
30.5
15.25
0.0
RPM
0.0
2.5
5.0
7.5
10.0
Seconds

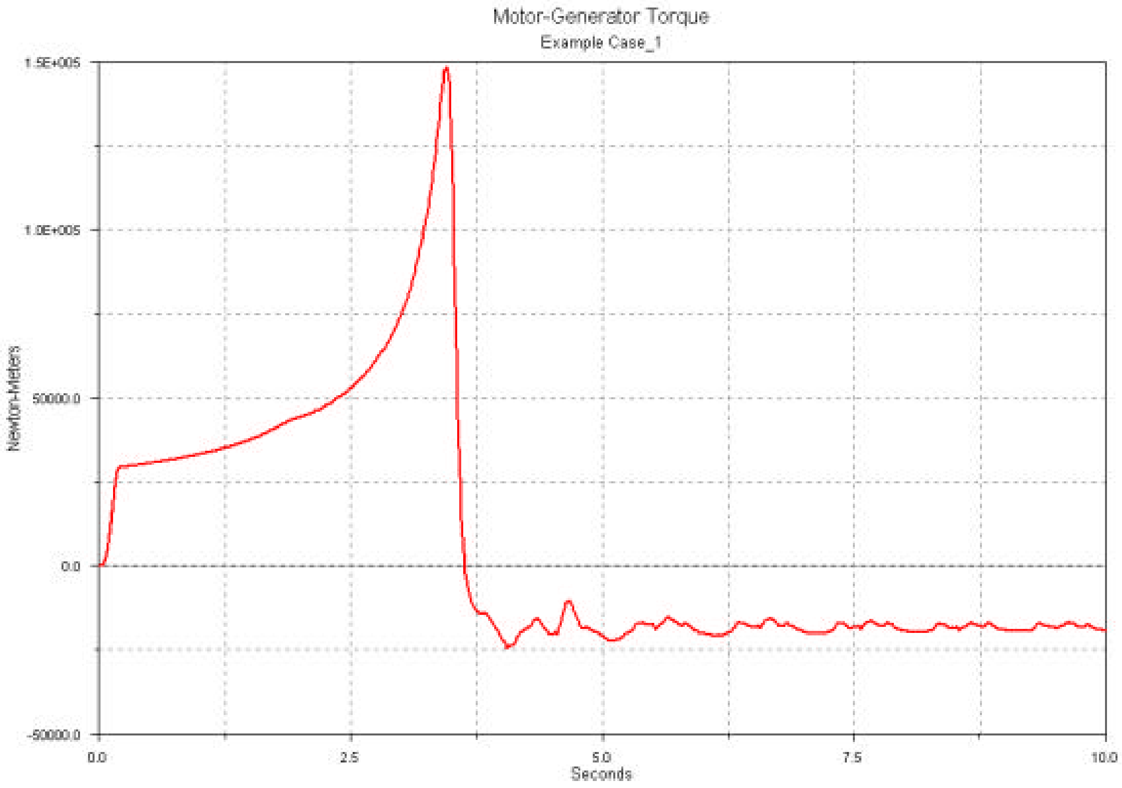

Nacelle Yaw
Example Case_1
6.35
0.0
-6.35
-12.7
-19.05
Degrees
0.0
2.5
5.0
7.5
10.0
Seconds


Blade Root Loads (forces)
Example Case_1
1.005E+005
67000.0
33500.0
0.0
-33500.0
Newtons
X
Y
Z
0.0
2.5
5.0
7.5
10.0
Seconds


Blade Root Loads (torques)
Example Case_1
1.2E+005
80000.0
40000.0
0.0
-40000.0
Newton-Meters
X
Y
Z
0.0
2.5
5.0
7.5
10.0
Seconds


Teetering Angle
Example Case_1
3.4
1.7
0.0
-1.7
-3.4
Degrees
0.0
2.5
5.0
7.5
10.0
Seconds


Blade 1 Azimuth Angle
Example Case_1
2805.0
1870.0
935.0
0.0
-935.0
Degrees
0.0
2.5
5.0
7.5
10.0
Seconds


Motor-Generator Torque
Example Case_1
1.5E+005
1.0E+005
50000.0
0.0
-50000.0
Newton-Meters
0.0
2.5
5.0
7.5
10.0
Seconds